



**UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

VINÍCIUS DOS REIS ALVES FERREIRA

**GERENCIAMENTO DA MALHA FERROVIÁRIA DE UMA USINA
SIDERÚRGICA: PROPOSTAS PARA A SIMULAÇÃO DE EVENTOS,
MODELAGEM DE PROCESSOS E O PLANEJAMENTO DE ROTAS BASEADO NA
METAHEURÍSTICA DE COLÔNIA DE FORMIGAS**

FORTALEZA

2019

VINÍCIUS DOS REIS ALVES FERREIRA

GERENCIAMENTO DA MALHA FERROVIÁRIA DE UMA USINA SIDERÚRGICA:
PROPOSTAS PARA A SIMULAÇÃO DE EVENTOS, MODELAGEM DE PROCESSOS E
O PLANEJAMENTO DE ROTAS BASEADO NA METAHEURÍSTICA DE COLÔNIA DE
FORMIGAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de Concentração: Automação e Controle.

Orientador: Prof. Dr. Arthur Plínio de Souza Braga.

FORTALEZA

2019

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

- F444g Ferreira, Vinícius dos Reis Alves.
Gerenciamento da Malha Ferroviária de uma Usina Siderúrgica : Propostas para a Simulação de Evendos, Modelagem de Processos, e o Planejamento de Rotas Baseado na Metaheurística de Colônia de Formigas / Vinícius dos Reis Alves Ferreira. – 2019.
112 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Fortaleza, 2019.
Orientação: Prof. Dr. Arthur Plínio de Souza Braga.
1. Otimização por colônia de formigas. 2. Modelo térmico. 3. Usina siderúrgica. 4. Roteamento de veículos. 5. Transporte de gusa. I. Título.

CDD 621.3

VINÍCIUS DOS REIS ALVES FERREIRA

GERENCIAMENTO DA MALHA FERROVIÁRIA DE UMA USINA SIDERÚRGICA:
PROPOSTAS PARA A SIMULAÇÃO DE EVENTOS, MODELAGEM DE PROCESSOS E
O PLANEJAMENTO DE ROTAS BASEADO NA METAHEURÍSTICA DE COLÔNIA DE
FORMIGAS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica. Área de Concentração: Automação e Controle.

Aprovada em: ___/___/_____.

BANCA EXAMINADORA

Prof. Dr. Arthur Plínio de Souza Braga (Orientador)
Departamento de Engenharia Elétrica
Universidade Federal do Ceará (UFC)

Prof. Dr. Wilkley Bezerra Correia (Membro Interno)
Departamento de Engenharia Elétrica
Universidade Federal do Ceará (UFC)

Prof. Dr. Anselmo Ramalho Pitombeira Neto (Membro Externo)
Departamento de Engenharia de Produção
Universidade Federal do Ceará (UFC)

A Deus.

A meus pais, esposa e filhos.

AGRADECIMENTOS

A Deus, pelas bênçãos e inúmeras oportunidades que proveu em minha vida.

À minha esposa Suellem e meu filho Miguel, pelo apoio incondicional e total compreensão nos momentos de ausência. Também a meu filho Henrique, que está chegando, por já tornar os dias mais especiais.

A meus pais Giovanni e Lindalva, pelo incentivo e por todos os esforços em fornecer uma educação de qualidade a mim e meu irmão.

Aos meus amigos e colegas de trabalho, em especial a meu amigo Lázaro, pelo apoio e incentivo em cursar o mestrado.

Ao Prof. Dr. Arthur Plínio de Souza Braga, pela excelente orientação.

Aos professores Prof. Dr. Anselmo Ramalho Pitombeira Neto, Prof. Dr. Fabrício Gonzalez Nogueira e Prof. Dr. Wilkley Bezerra Correia pelo tempo e pelas valiosas colaborações.

E a todos que contribuíram de alguma forma para a realização deste sonho.

*“Quanto mais aumenta nosso conhecimento,
mais evidente fica nossa ignorância.”*

John F. Kennedy

RESUMO

A otimização de processos e a redução de custos são tópicos interligados fundamentais para a sobrevivência de qualquer empresa. Cada dia mais recursos são investidos em automação e na melhoria contínua dos processos industriais com foco em excelência operacional. Com base nisso, este trabalho propõe a redução de custos relacionados às perdas do processo de distribuição de ferro gusa em uma usina siderúrgica integrada. Neste tipo de siderúrgica, o ferro líquido produzido no alto forno é transportado para a aciaria, onde será transformado em aço. Este transporte é realizado na grande maioria das siderúrgicas através de veículos sobre trilhos como locomotivas e carros torpedo. A distância percorrida e o tempo gasto neste processo têm impacto direto em perda de energia térmica do ferro gusa para o ambiente, o que implica em maiores custos para a produção do aço, além dos custos relacionados ao transporte. Este trabalho propõe um modelo baseado no problema de roteamento de veículos para a otimização do gerenciamento e transporte do ferro gusa em uma usina siderúrgica. Também é criado um modelo térmico para estimar a temperatura do metal e do carro torpedo durante o transporte entre pontos onde não é possível realizar a medição de temperatura. Para a resolução do problema foi adotada a metaheurística de Otimização por Colônia de Formigas aplicada no roteamento de veículos e na redução da variação de temperatura do carro torpedo, com o objetivo de diminuir as perdas térmicas e também o desgaste do material refratário que ocorre devido a estas variações. Os modelos foram testados através de um simulador desenvolvido e os resultados comparados com dados operacionais de uma organização inserida no setor siderúrgico nacional que tem como principal produto a fabricação de placas de aço. Dados relacionados ao desempenho e aplicabilidade do algoritmo em ambiente de produção também são discutidos. Como principais resultados alcançados pelo algoritmo pode-se destacar uma redução de 17% na distância total percorrida na distribuição do ferro gusa, além de uma redução das quedas de temperatura de 21 e 25% para os torpedos vazios e carregados, respectivamente. Estes resultados representaram um aumento aproximado de 40°C na temperatura média do ferro gusa entregue na aciaria frente aos resultados obtidos através das tomadas de ações de operadores experientes.

Palavras-chave: Otimização por colônia de formigas. Modelo térmico. Usina siderúrgica. Roteamento de veículos. Transporte de gusa.

ABSTRACT

Process optimization and cost reduction are key interrelated topics for the survival of any business. More and more resources are being invested in automation and continuous improvement of industrial processes focused on operational excellence. Based on this, the present work proposes the reduction of costs related to the losses of the pig iron transportation process in an integrated steel mill. In this type of steel mill, the pig iron produced in blast furnace is transported to steelmaking plant, where it will be transformed into steel. This transport is carried out on the most of steel mills by rail vehicles such as locomotives and torpedo ladle cars. The distance traveled and the time spent in this process have a direct impact on the loss of pig iron thermal energy to the environment, which implies higher costs for steel production, besides transportation costs. This work proposes a model based on the vehicle routing problem for the optimization of railway traffic management and pig iron transportation in a steel mill. A thermal model is also created to estimate the temperature of the metal and torpedo ladle car during transport between points where temperature measurement is not possible. To solve the problem, it was chose a metaheuristic based on ant colony optimization applied to vehicle routing and to reducing temperature variation of the torpedo ladle car, which seeks to reduce temperature losses and also the wear of refractory material that occurs due to these variations. The models were tested using a simulator developed and the results compared with operational data of a Brazilian steel mill whose main product is the manufacture of steel slabs. Data related to the performance and applicability of the algorithm in production environment are also discussed. The main results achieved by the algorithm include a 17% reduction in the total distance traveled, in addition to a 21 and 25% reduction in thermal losses for empty and charged torpedo ladle cars, respectively, which represented an approximate 40°C increase in the average temperature of the pig iron delivered to the steelmaking plant against the results obtained through the actions taken by experienced operators.

Keywords: Ant Colony Optimization. Thermal model. Steel mill. The Vehicle Routing Problem. Pig Iron Transportation.

LISTA DE FIGURAS

Figura 1 – Processo macro de um alto forno	20
Figura 2 – Processo macro de uma aciaria	23
Figura 3 – Carro torpedo	25
Figura 4 – Tempo médio para cada etapa dos ciclos de um torpedo	26
Figura 5 – Etapas de transporte do ferro gusa	27
Figura 6 – Malha ferroviária simplificada de uma usina siderúrgica	28
Figura 7 – Comportamento das formigas	41
Figura 8 – Fluxograma do processo de otimização de associação das requisições	47
Figura 9 – Fluxograma do algoritmo de construção do grafo das soluções	48
Figura 10 – Grafo criado para a solução final do problema	49
Figura 11 – Disposição de torpedos tracionados por uma locomotiva	50
Figura 12 – Fluxograma do algoritmo de colônia de formigas	51
Figura 13 – Grafo que representa a malha ferroviária de uma usina siderúrgica	61
Figura 14 – Interface gráfica do Simulador	62
Figura 15 – Interface gráfica do Simulador durante simulação	63
Figura 16 – Classes do simulador	65
Figura 17 – Fluxograma do Simulador	67

LISTA DE GRÁFICOS

Gráfico 1 – Modelo térmico do TLC vazio: variação de temperatura x tempo.....	53
Gráfico 2 – Modelo térmico do TLC vazio: variação de temp. por hora x temp. de descarga.	53
Gráfico 3 – Modelo térmico do TLC cheio: variação de temperatura x temperatura do gusa .	55
Gráfico 4 – Modelo térmico do TLC cheio: variação de temperatura x temperatura do TLC .	56
Gráfico 5 – Modelo térmico do TLC cheio: variação de temperatura x tempo.....	57
Gráfico 6 – Validação do modelo térmico de torpedo carregado	58
Gráfico 7 – Distribuição do peso de gusa por corrida do alto forno	69
Gráfico 8 – Distribuição do peso de gusa por corrida da aciaria.....	70
Gráfico 9 – Distribuição do intervalo entre corridas do alto forno	70
Gráfico 10 – Distribuição do intervalo entre corridas da aciaria.....	71
Gráfico 11 – Distribuição da temp. média do ferro gusa em uma corrida do alto forno.....	72
Gráfico 12 – Simulação 1: Distribuição de utilização das locomotivas ao longo do tempo	80
Gráfico 13 – Simulação 3: Queda de temperatura (a) ΔT_{Fe} ; (b) ΔT_{TLC}	86
Gráfico 14 – Simulação 3: Indicadores de processamento.....	87

LISTA DE TABELAS

Tabela 1 – Composição típica do ferro-gusa e aço.....	22
Tabela 2 – Constantes adotadas na simulação	72
Tabela 3 – Parâmetros adotados para o algoritmo	74
Tabela 4 – Indicadores de desempenho de processo	76
Tabela 5 – Indicadores de desempenho de algoritmo.....	77
Tabela 6 – Simulação 1: Resultados de desempenho de processo	78
Tabela 7 – Simulação 1: Taxa de utilização da locomotiva e desempenho do algoritmo	79
Tabela 8 – Simulação 2: Custos de temperatura.....	81
Tabela 9 – Simulação 2: Resultados	82
Tabela 10 – Simulação 2: Desvio Padrão para as simulações	83
Tabela 11 – Simulação 2: Classificação	83
Tabela 12 – Simulação 3: Custo deslocamento e taxa de utilização locomotivas.....	85
Tabela 13 – Simulação 3: Queda de temperatura	85
Tabela 14 – Simulação 3: Indicadores de processamento	86

LISTA DE ABREVIATURAS E SIGLAS

ACO	- <i>Ant Colony Optimization</i>
ACS	- <i>Ant Colony System</i>
AS	- <i>Ant System</i>
ASelite	- <i>Elit Ant System</i>
ASrank	- <i>Rank based Ant System</i>
BF	- <i>Blast Furnace (Alto Forno)</i>
BOF	- <i>Basic Oxygen Furnace</i>
CLP	- <i>Controlador Lógico Programável</i>
DARP	- <i>Dial-a-Ride Problem</i>
DMDVRP	- <i>Dynamic Multi-Depot Vehicle Routing Problem</i>
DPDVRP	- <i>Dynamic Pickup and Delivery Vehicle Routing Problem</i>
DRI	- <i>Direct Reduced Iron</i>
DVRP	- <i>Dynamic Vehicle Routing Problem</i>
EAF	- <i>Electric Arc Furnace</i>
EIE	- <i>Electronic Interlocking Equipment</i>
GPS	- <i>Global Positioning System</i>
GVRP-MDMPDR	- <i>Generalized Vehicle Routing Problem for Multiple Depots with Multiple Pickup and Delivery requests</i>
HMI	- <i>Human Machine Interface</i>
KR	- <i>Kanbara Reactor</i>
LD	- <i>Ladle Furnace</i>
LIFO	- <i>Last In First Out</i>
MDVRP	- <i>Multi-depot Vehicle Routing Problem</i>
MMAS	- <i>Min-Max Ant System</i>
PDVRP	- <i>Pickup and Delivery Vehicle Routing Problem</i>
PSO	- <i>Particle Swarm Optimization</i>
RFID	- <i>Radio-Frequency IDentification</i>
RH	- <i>Ruhrstahl Heraeus</i>
RMSE	- <i>Root Mean Square Error</i>
SD	- <i>Slag Disposal Pit</i>
SM	- <i>Steel Making (Aciaria)</i>

TLC - *Torpedo Ladle Car*
TSP - *Traveling Salesman Problem*
VRP - *Vehicle Routing Problem*
VRPTW - *Vehicle Routing Problem with Time Windows*

LISTA DE SÍMBOLOS

η	- Atratividade da aresta
C	- Capacidade do veículo
τ	- Concentração de feromônio
τ_0	- Concentração mínima de feromônio por aresta
G	- Conjunto de depósitos
P	- Conjunto de pontos de coleta
D	- Conjunto de pontos de entrega
V	- Conjunto de todos os nós
N	- Conjunto de todos os pontos de <i>pickup</i> e <i>delivery</i>
K	- Conjunto de veículos
A	- Conjunto dos arcos
Q	- Constante de atualização de feromônio
f	- Custo fixo do veículo caso ele seja utilizado
d	- Distância percorrida
GB	- Giga bytes
GHz	- Giga hertz
GJ	- Giga joules
$^{\circ}C$	- Graus celsius
h	- Hora
m	- Metro
min	- Minuto
K	- Número de formigas
n	- Número de iterações
n_{TLC}	- Número de torpedos disponíveis
n_{Loc}	- Número de veículos
q_0	- Parâmetro de escolha entre regra padrão do AS e a regra de maior atratividade
β	- Parâmetro de influência da qualidade da aresta
α	- Parâmetro de influência de feromônio
%	- Porcentagem
l	- Quantidade de carros torpedo a serem carregados
u	- Quantidade de carros torpedo a serem descarregados

ΔT_{Fe}	- Queda de temperatura do gusa
ΔT_{TLC}	- Queda de temperatura do torpedo
km	- Quilômetros
s	- Segundos
ρ_1	- Taxa de evaporação global de feromônios
ρ_2	- Taxa de evaporação local de feromônios
r_{Loc}	- Taxa de ocupação da locomotiva
T	- Temperatura
T_{Fe}^{SM}	- Temperatura do gusa na descarga
T_{TLC}^{SM}	- Temperatura do torpedo na descarga
T_{TLC}^{BF}	- Temperatura do torpedo no enchimento
T_{Fe}^{BF}	- Temperatura gusa no enchimento
t	- Tempo
t_{proc}^{max}	- Tempo máximo de processamento
\bar{t}_{proc}	- Tempo médio de processamento
t_a	- Tempo total de atraso de corridas
ton	- Toneladas
x	- Variável binária de decisão
v	- Velocidade
v_t	- Velocidade média da locomotiva
v_d	- Velocidade média de descarga do torpedo
v_c	- Velocidade média enchimento de torpedo

SUMÁRIO

1 INTRODUÇÃO	14
1.1 Motivação da Pesquisa	15
1.2 Objetivos Gerais e Específicos.....	15
1.3 Metodologia	16
1.4 Produção Bibliográfica da Pesquisa	17
1.5. Organização da Dissertação	17
2 PRODUÇÃO DO AÇO: PROCESSOS ENVOLVIDOS E A RELEVÂNCIA DA EFICIÊNCIA NO TRANSPORTE DO FERRO-GUSA	19
2.1 Produção de Ferro-Gusa.....	19
2.2 Produção de Aço	21
2.3 Transporte do Ferro-Gusa	25
2.3.1 Ciclo do Torpedo e Perdas do Processo.....	25
2.3.2 A Malha Ferroviária e a Dinâmica Operacional	28
2.4 Importância do Processo de Distribuição de Gusa.....	31
3 PROBLEMA DE ROTEAMENTO DE VEÍCULOS	32
3.1 Modelo Matemático para o Problema de Roteamento.....	35
3.1.1 Conjuntos.....	36
3.1.2 Parâmetros	36
3.1.3 Variáveis de Decisão.....	37
3.1.4 Modelo do Problema	37
3.2 Otimização por Colônia de Formigas	39
3.2.1 Ant System (AS).....	40
3.2.2 Ant Colony System (ACS)	43
3.3 Metaheurística de Colônia de Formigas Aplicada ao Roteamento de Veículos	45
4 GERENCIAMENTO POR COLÔNIA DE FORMIGAS DA MALHA FERROVIÁRIA DE UMA USINA SIDERÚRGICA	46
4.1 Algoritmo Proposto.....	46
4.2 Modelos de Temperatura	51
4.2.1 Modelo do Torpedo Vazio	52
4.2.2 Modelo do Torpedo Carregado	54
4.2.3 Validação dos Modelos Térmicos	58
4.3 Validação e Testes do Algoritmo	59
5 O SIMULADOR DE EVENTOS DA MALHA FERROVIÁRIA DE UMA USINA SIDERÚRGICA	60
5.1 Modelamento da Malha Ferroviária.....	60
5.2 Interface do Simulador	61

5.3 Estrutura do Simulador	64
5.4 Simulações do Processo	68
5.5 Informações Adicionais Sobre as Simulações	73
6 RESULTADOS E ANÁLISES	74
6.1 Indicadores de Desempenho	75
6.2 Simulação 1 – Operador <i>versus</i> Algoritmo	77
6.3 Simulação 2 – Variação no Custo de Temperatura.....	80
6.4 Simulação 3 – Testando Cenários de Intervenção	84
7 CONSIDERAÇÕES FINAIS.....	89
7.1 Trabalhos futuros.....	90
REFERÊNCIAS	91
APÊNDICE A – ARTIGO CBSF 2018.....	95

1 INTRODUÇÃO

Em uma usina siderúrgica, um dos primeiros passos para produção do aço consiste na obtenção do ferro gusa (ferro líquido) que é produzido na planta do alto forno. Neste processo, minério de ferro e combustíveis/redutores (coque e carvão) são transformados em gusa, escória e gás. O ferro gusa é então transportado para a aciaria, onde ocorre o refino deste em aço a partir da remoção de elementos como enxofre, silício, carbono, manganês e fósforo (GEERDES, 2007). O gás produzido é utilizado na geração de energia elétrica e a escória torna-se insumo para outras indústrias, como a indústria cimenteira.

Para a produção do aço, o ferro gusa precisa ser transportado do alto forno para a aciaria. Este transporte é realizado com a utilização de veículos sobre trilhos, como locomotivas e carros torpedo. O tempo gasto nesse percurso influencia diretamente na perda de energia térmica para o meio ambiente, refletindo no maior consumo de redutor no alto forno ou na redução da carga metálica de sucata acrescida na aciaria (MAGNANI, 2002), o que implica em maiores gastos e um conseqüente aumento no custo de produção do aço.

A malha ferroviária de uma usina siderúrgica é composta, basicamente, pelos seguintes elementos: (1) postos de abastecimento de carro torpedo (alto forno); (2) postos de descarga de ferro gusa (aciaria); (3) aquecedores (local destinado ao aquecimento do carro torpedo); (4) postos de basculamento de emergência; (5) postos de limpeza de carro torpedo; e (6) oficina de carro torpedo/locomotivas.

A logística e o gerenciamento deste tráfego requer a garantia de disponibilidade de carros torpedo vazios para o recebimento de ferro-gusa nas corridas do alto forno e a disponibilização de carros torpedo cheios para suprir as necessidades da aciaria. Também é preciso garantir saldo de gusa suficiente para atender as demandas da aciaria em casos de problemas operacionais no alto forno, e uma baixa perda de energia térmica do gusa. Além disto, é necessário manter a variação de temperatura do torpedo dentro de determinados limites, para assegurar maior vida útil do material refratário que reveste o interior do veículo.

Frente tais necessidades, a decisão de qual torpedo enviar para atender uma demanda do alto forno ou aciaria torna-se complexa devido à grande quantidade de variáveis a serem analisadas, como: a temperatura do torpedo vazio, temperatura do gusa, quantidade de gusa no torpedo, distribuição dos veículos na malha ferroviária, distância do veículo até o destino e quantidade de ciclos do torpedo.

Dada a complexidade do gerenciamento dos veículos e do tráfego, a otimização deste processo pode resultar em aumento da disponibilidade das áreas operacionais, redução

dos custos de produção, redução dos gastos com manutenção e consequente diminuição de impactos ambientais gerados pelo descarte de material refratário.

1.1 Motivação da Pesquisa

No disputado cenário econômico atual, a redução de custos e a otimização de processos são fundamentais para a sobrevivência de qualquer empresa. Este trabalho busca, através da otimização, a redução de custos relacionados às perdas no processo de distribuição de ferro gusa em uma usina siderúrgica integrada.

Devido à grande quantidade de variáveis a serem analisadas, a decisão de qual torpedo enviar para atender uma determinada demanda e o gerenciamento do tráfego da malha ferroviária tornam-se atividades um tanto complexas, o que dificulta a tomada de decisão por parte dos operadores. Além disso, a otimização deste processo pode resultar em aumento de disponibilidade, redução dos custos de produção e de manutenção, além da diminuição de impactos ambientais gerados pelo descarte de material refratário.

Uma das possibilidades para otimizar o gerenciamento de torpedos na malha ferroviária é a utilização de inteligência artificial para auxiliar a tomada de decisão do operador, bem como definir o melhor percurso a ser adotado. O problema de roteamento de veículos (Vehicle Routing Problem - VRP) é um dos principais problemas de otimização combinatória e consiste em um determinado número de consumidores que solicitam a entrega de certa quantidade de material ou produtos. Nele, cada veículo realiza uma rota partindo de um ou mais pontos denominados depósitos, percorrendo os nós que representam os consumidores de forma que todas as demandas sejam atendidas e o custo total de transporte seja minimizado. Cada veículo possui uma capacidade de carga que deve ser respeitada, além da distância total percorrida que pode ser limitada.

1.2 Objetivos Gerais e Específicos

Com base na importância do processo de distribuição de gusa e o impacto deste nos custos de produção do aço, este trabalho tem como objetivo principal a criação de um simulador que, associado a uma metaheurística, possibilite a validação e testes dos modelos propostos, bem como a comparação dos resultados simulados com dados reais de processo. A metaheurística deve realizar a otimização das tomadas de decisão referentes ao gerenciamento

de carros torpedo e tráfego da malha ferroviária de uma usina siderúrgica. Além disso, lista-se como objetivos específicos a serem alcançados na solução proposta neste trabalho:

- Diminuir a variação de temperatura do carro torpedo e do ferro gusa. Para isso faz-se necessário a elaboração de um modelo térmico para o carro torpedo, a fim de estimar a temperatura do mesmo em pontos em que não é possível realizar a medição.
- Analisar e detalhar o processo de distribuição de gusa, verificando os requisitos e restrições inerentes ao processo.

1.3 Metodologia

A organização que forneceu dados para a presente pesquisa, base para o problema estudado, está inserida no setor siderúrgico nacional e tem como atividade a fabricação de placas de aço, cuja grande maioria é destinada ao mercado externo. Metodologicamente, os dados para o presente trabalho provêm de pesquisa do tipo documental, pesquisa de campo e simulações.

Aproximadamente seis meses de dados foram colhidos e são utilizados para criação de modelos e padrões que caracterizam o processo de distribuição de gusa e os processos adjacentes: produção de ferro gusa (realizado no alto forno) e produção de aço (aciaria). Através destes modelos, um simulador foi desenvolvido para emular as condições de processo e permitir a criação de diferentes cenários de simulação. O simulador pode ser alimentado com dados históricos do processo referentes à um determinado período, o que permite a comparação dos resultados obtidos pelo algoritmo de otimização frente às mesmas condições enfrentadas por um operador.

Dessa forma, os resultados obtidos pelo simulador são comparados com os resultados operacionais referentes ao mesmo período e condições de operação que foram simulados. Os principais indicadores de desempenho adotados estão relacionados à distância total percorrida, à variação de temperatura média dos carros torpedo e à queda de temperatura do ferro gusa durante o transporte. Além disso, indicadores de desempenho do algoritmo foram criados para análise de viabilidade de utilização deste no ambiente fabril.

1.4 Produção Bibliográfica da Pesquisa

Parte dos resultados desta pesquisa foram publicados no artigo “*Um Sistema Fuzzy para Ajuste Dinâmico dos Parâmetros de um Sistema de Colônia de Formigas Aplicado ao Controle Ferroviário de uma Usina Siderúrgica*” apresentado no **V Congresso Brasileiro de Sistemas Fuzzy – CBSF**. O trabalho apresenta um algoritmo híbrido aplicado na otimização de rotas da malha ferroviária de uma usina siderúrgica. O algoritmo consiste de um controlador Fuzzy responsável pelo ajuste dinâmico dos parâmetros da metaheurística baseada em Colônia de Formigas, com o intuito de controlar a exploração de novas soluções e a intensificação de buscas no espaço de soluções através de soluções já conhecidas.

1.5. Organização da Dissertação

Este trabalho está estruturado em sete capítulos e os temas abordados em cada um são detalhados a seguir:

O **Capítulo 1** traz uma breve introdução e contextualização do tema desta dissertação, com destaque à problematização, principais objetivos e possíveis soluções que serão abordadas ao longo deste trabalho.

O **Capítulo 2** detalha parte dos processos de produção de ferro gusa e aço de uma usina siderúrgica, bem como o processo de transporte e distribuição do mesmo. Além disso traz informações importantes sobre a dinâmica de operação e integração realizada pela equipe de transporte de gusa com as áreas alto forno e aciaria.

No **Capítulo 3** foi realizada uma revisão do referencial teórico fundamentado em duas linhas. A primeira relativa ao problema de roteamento de veículos, suas aplicações e principais variações do problema. A segunda linha está vinculada aos principais métodos de resolução do problema de roteamento de veículos, em especial a aplicação de otimização por colônia de formigas. Junto ao problema de roteamento de veículos, o modelo do problema de otimização da malha ferroviária é apresentado neste capítulo.

Em seguida, no **Capítulo 4**, é apresentado o algoritmo proposto para a solução do problema de roteamento da malha ferroviária. Além do algoritmo, também são apresentados os modelos de temperatura utilizados para a estimação da temperatura do carro torpedo e do ferro gusa durante o transporte.

O **Capítulo 5** aborda o simulador desenvolvido para a simulação das condições da malha ferroviária e do processo de transporte de gusa. As principais variáveis do processo são estruturadas e mostradas a fim de representar a dinâmica da produção do aço. Junto ao simulador é apresentado também o grafo que representa a malha ferroviária da usina siderúrgica.

O **Capítulo 6** traz o detalhamento das premissas adotadas para simulação, onde também são abordados os principais resultados. Um comparativo é realizado entre os resultados obtidos com o algoritmo descrito no Capítulo 4 e os resultados oriundos da operação cotidiana no ambiente industrial.

Por fim, o **Capítulo 7** apresenta as considerações finais com um resumo do trabalho desenvolvido e seus impactos nas comunidades científicas e industrial. Além disso, propõe-se sugestões de trabalhos futuros para continuidade da pesquisa.

O **Apêndice A** mostra o artigo apresentado no **V Congresso Brasileiro de Sistemas Fuzzy – CBSF** que traz resultados interessantes do uso de otimização por Colônia de Formigas aplicado à otimização de rotas da malha ferroviária de uma usina siderúrgica.

2 PRODUÇÃO DO AÇO: PROCESSOS ENVOLVIDOS E A RELEVÂNCIA DA EFICIÊNCIA NO TRANSPORTE DO FERRO-GUSA

O aço é uma das mais importantes matérias primas no mundo hoje. Em 2017 a produção mundial de aço foi de 1690 milhões de toneladas, um crescimento de 3,9% comparado com 2016 (*World Steel Association*, 2018). O alto forno é o processo mais utilizado na produção de ferro-gusa, matéria prima utilizada para a produção do aço. Aproximadamente 70% do ferro líquido utilizado na produção de aço é originado do processo de alto forno e transportado para a aciaria em carros torpedo ou panelas. Para a melhor compreensão do processo de transporte de ferro-gusa, faz-se necessário um entendimento básico dos processos de produção de ferro-gusa e aço, bem como uma visão geral da dinâmica de operação do alto forno e da aciaria.

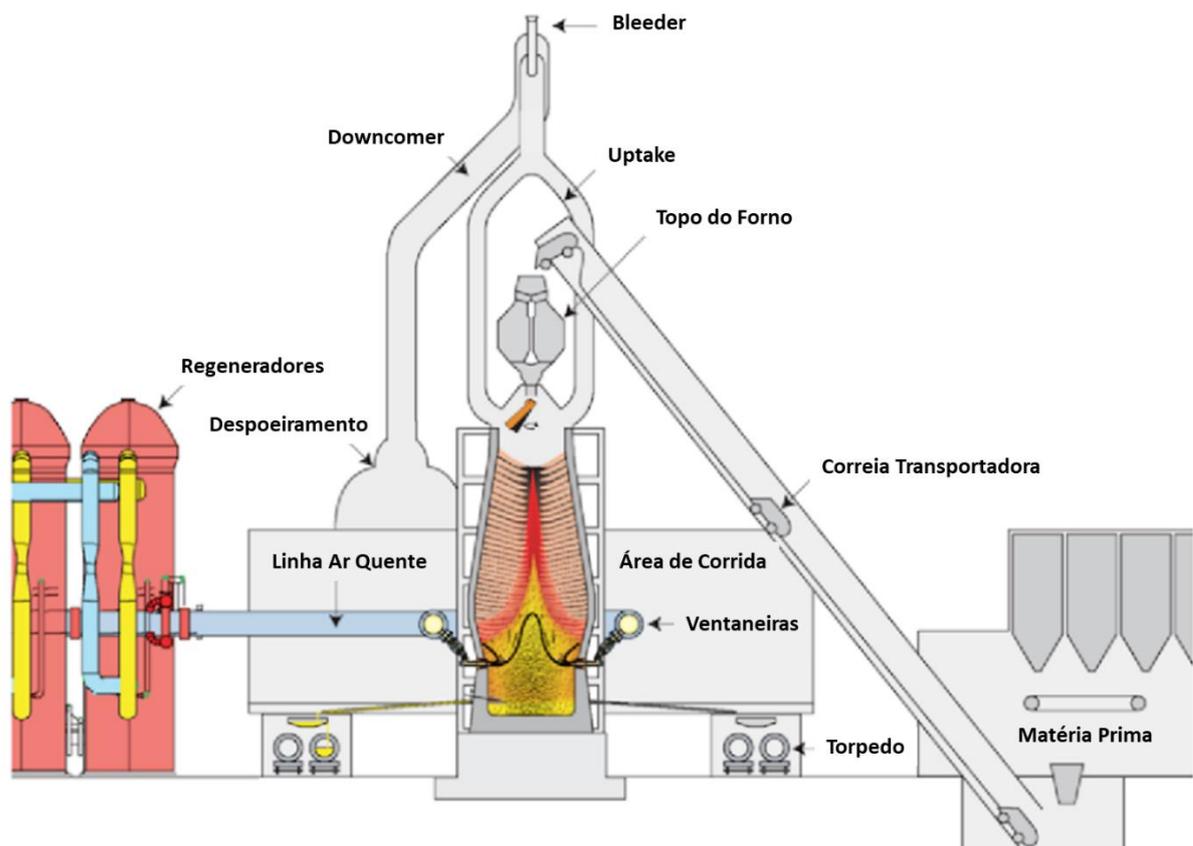
Para a descrição dos processos envolvidos com o tema deste trabalho, o restante deste capítulo está organizado conforme se segue. As Seções 2.1 e 2.2 detalham os processos de produção de ferro-gusa no alto forno e de produção e aço na aciaria, que são processos adjacentes à distribuição de gusa e, por isso, impactam diretamente no *modus operandi* deste. A Seção 2.3 apresenta as principais características do processo de distribuição de gusa. E a Seção 2.4 encerra o capítulo destacando a relevância da otimização das rotas na eficiência da produção do aço – foco desta pesquisa.

2.1 Produção de Ferro-Gusa

No alto forno, minério de ferro e combustíveis/redutores (coque e carvão) são transformados em gusa, escória e gás. O minério e o coque são carregados pelo topo do forno em camadas alternadas, mantendo o alto forno cheio destes materiais utilizados no processo de produção de gusa. O ar quente com temperatura de até 1300°C é soprado, juntamente com combustível auxiliar (carvão pulverizado) para dentro do alto forno pelas ventaneiras, uma peça de cobre refrigerada à água que pode ter até 1m de comprimento. As ventaneiras estão dispostas lado a lado no mesmo nível e são interligadas por um anel de vento. A quantidade de ventaneiras depende do diâmetro do alto forno, podendo variar de 30 a 40 ventaneiras para alto fornos maiores. O ar soprado queima os combustíveis presentes no forno gerando um gás com alta temperatura de chama (2100 a 2300°C) através da conversão do oxigênio em monóxido de carbono. Este gás ascende pelo forno aquecendo os materiais nas regiões superiores e causando a fusão do minério de ferro, que resulta em gusa e escória e se mantém na forma líquida no cadinho.

Um alto forno moderno possui entre 2 e 4 furos de gusa e um número correspondente de canais. Para cada canal/furo de gusa existe uma calha basculante e duas linhas férreas por onde chegam os torpedos para recebimento de gusa, como pode ser visto na Figura 1. A calha basculante tem a função de possibilitar a seleção de qual torpedo irá receber o ferro-gusa durante um vazamento. Isso se faz necessário pois, durante uma corrida, a quantidade de gusa produzido é superior à capacidade de um único carro torpedo e, portanto, é preciso efetuar a troca de carros torpedo durante o processo de vazamento. A separação da escória do ferro-gusa é realizada pela diferença de densidade durante a descida dos líquidos pelo canal de corrida, de forma que gusa é carregado no carro torpedo, e a escória, menos densa, é desviada e passa por um processo de granulação ou resfriamento para, em seguida, ser destinada como subproduto para outras indústrias ou processos.

Figura 1 – Processo macro de um alto forno



Fonte: Geerds *et al.* (2015)

Um alto forno é vazado de 8 a 14 vezes por dia através do furo de corrida. Cada vazamento é chamado de corrida e tem duração média de 1,5 à 3h. Durante uma corrida, o forno produz uma parte considerável de seu volume de trabalho. Para o alto forno, objeto de estudo deste trabalho, tem-se uma quantidade média de vazamento de 940 toneladas de ferro gusa por

corrida, o que corresponde à aproximadamente 3,1 carros torpedo, uma vez que cada carro torpedo possui capacidade de carga de 300 toneladas de gusa.

É através do vazamento que o nível do líquido no cadinho pode ser controlado, e esta operação possui dois efeitos principais no processo do alto forno: (1) os líquidos no cadinho afetam a descida de carga, em que quanto maior o nível maior será a força que empurrará a carga em sentido ascendente; (2) se a escória alcançar o nível das ventaneiras e não puder ser drenada para o cadinho, o fluxo de gás será severamente afetado, o que pode resultar numa redução deficiente da carga e desta forma na redução de temperatura do forno (GEERDS *et al.*, 2007). Por isso é de suma importância o atendimento das requisições de carros torpedo do alto forno para garantir que a programação das corridas seja mantida e que o nível dos líquidos no cadinho seja controlado preferivelmente em um nível constante e baixo. Além disso, é importante ressaltar que o objetivo das operações de vazamento é conseguir extrair o líquido do cadinho sem interferência com o processo do alto forno.

2.2 Produção de Aço

A produção de aço se inicia com o recebimento do ferro-gusa na aciaria, quando o metal líquido entregue por um carro torpedo é transferido para uma panela de gusa. Cada panela representa uma corrida de aciaria e possui capacidade similar à de um carro torpedo: aproximadamente 300t.

Segundo Mazumdar e Evans (2011), o processo de produção de aço pode ser classificado em três processos parciais distintos: refino primário, refino secundário e lingotamento contínuo. No refino primário é produzido o aço bruto que posteriormente é refinado em termos de composição e pureza através de uma série de processos secundários (refino secundário). Por fim, o aço é transformado em produtos sólidos através do lingotamento contínuo. Estes produtos são principalmente placas, lingotes ou tarugos.

Além destes três processos, uma fase de pré-tratamento é necessária para a remoção de algumas impurezas do ferro-gusa, como silício, enxofre e fósforo. Em grande parte das siderúrgicas norte-americanas, o ferro-gusa produzido no alto forno possui concentração de enxofre entre 0,040 e 0,070%, enquanto o gusa carregado nos convertedores deve ter concentração entre 0,001 e 0,010% para se conseguir um aço de boa qualidade (NIIR, 2010). Algumas destas impurezas poderiam ser removidas durante o refino primário, entretanto seria necessário um aumento do tempo de sopro o que gera maior desgaste do refratário do convertedor. Além disso, os processos utilizados no pré-tratamento implicam em menor custo.

Dois processos principais são utilizados para tratamento do ferro-gusa: (1) HMPS – *Hot Metal Pretreatment Station* e (2) KR – *Kambara Reactor*. O primeiro realiza principalmente a remoção de silício e fósforo através da dissolução de reagentes baseados em calcário, às vezes misturados com minério de ferro ou finos de sinter entregues com uma injeção de oxigênio para diminuir o impacto da perda de energia térmica no processo. Já o segundo realiza remoção do enxofre do gusa através da injeção de cal, fluoreto de cálcio e óxido de alumínio através de uma lança e com a utilização de um gás de transporte, geralmente nitrogênio. Além do sistema de injeção, o KR é equipado de um rotor e algumas pás, responsáveis por agitar o metal líquido causando uma turbulência que tem o objetivo de aumentar a eficiência do processo. O aumento da eficiência deve-se à redução do tamanho das bolhas do gás de transporte que aumenta o tempo de residência dos materiais reagentes no gusa (SCHRAMA; BERG, 2014).

Existem basicamente duas rotas referentes ao refino primário utilizadas para a produção de aço: (1) BOF – *Basic Oxygen Furnace*; e (2) EAF – *Electric Arc Furnace*. No primeiro, também denominado convertedor, o ferro gusa recebido do alto forno é refinado através de um ambiente oxidante formado pela injeção de oxigênio a uma velocidade supersônica no aço fundido através de uma lança (GHOSH; CHATTERJEE, 2008). Durante uma corrida de convertedor, ferro gusa, sucata metálica, cal, minério de ferro, dentre outros aditivos, são carregados no convertedor (BOF). Com a injeção de oxigênio, as impurezas são oxidadas formando a escória e esta reação gera uma grande quantidade de calor. Este calor é utilizado para elevação da temperatura do aço bruto, e em conjunto com a temperatura do ferro-gusa recebido dita a quantidade de sucata metálica que pode ser adicionada ao processo.

Na Tabela 1 é possível ver as faixas de temperatura de cada fase da produção de aço. Ao final do processo o aço bruto é destinado à uma panela de recebimento de aço que segue para o refino secundário e a escória é separada em um pote de escória.

Tabela 1 – Composição típica do ferro-gusa e aço

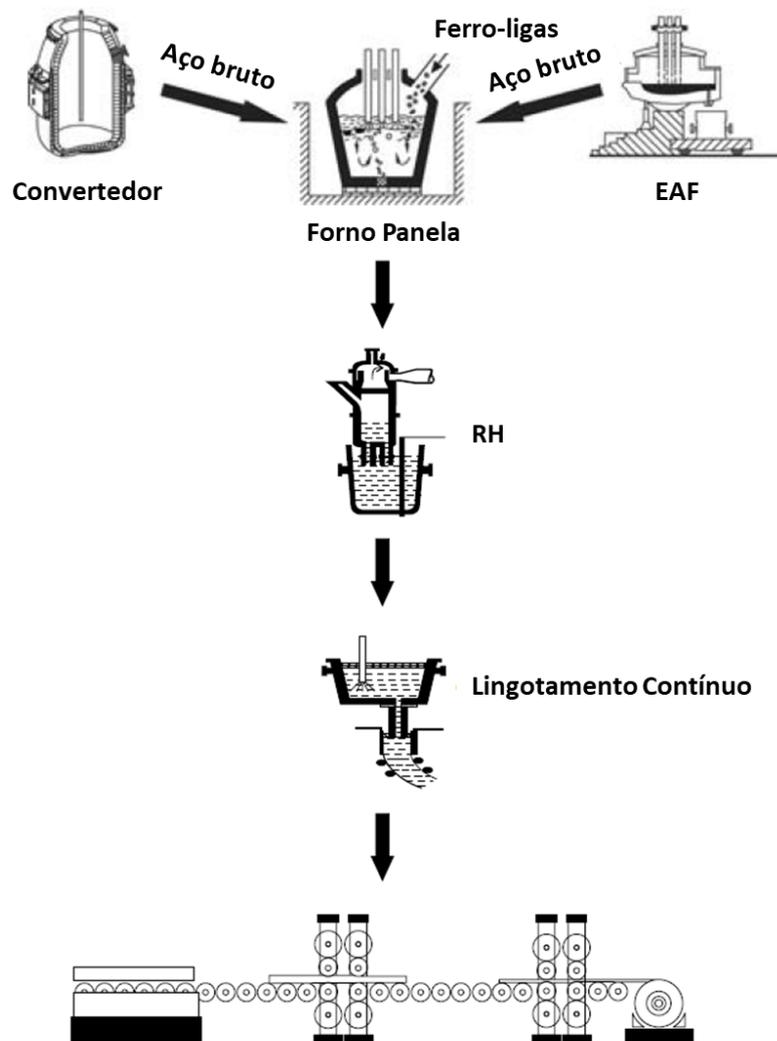
Produto	Processo	Composição Típica	Temperatura (°C)
Ferro-gusa	Alto Forno	[%C] = 4.5; [%S] = 0.05; [%P] = 0.15; [%Si] = 1.5; [%Mn] = 1.0; [O] < 1 ppm	1300-1400
Aço bruto	BOF	[%C] = 0.05; [%S] = 0.02; [%P] = 0.008; [%Si] = 0.1; [%Mn] = 0.2; [O] = 400 ppm; [N] = 40 ppm	1550-1650
Aço acabado	Refino Secundário	[%C] = 0.05; [%S] = 0.008; [%P] = 0.008; [%Si] = 0.1; [%Mn] = 0.2; [O] < 20 ppm; [N] < 10 ppm; [H] < 1 ppm	1550-1580

Fonte: Adaptado de Mazumdar e Evans. (2011)

Já na segunda rota, o forno elétrico à arco – EAF é carregado com sucata de aço e ferro de redução direta (DRI – *Direct Reduced Iron*), ambos em estado sólido. O aquecimento e fusão destes materiais é realizada através do calor gerado pelo arco elétrico criado pelos eletrodos de grafite que dão nome ao processo. O total de energia gasta na produção de 1 tonelada de aço é de aproximadamente 6,6GJ, considerando energia elétrica, queimadores e energia química. O EAF não será profundamente abordado neste trabalho pois não utiliza ferro-gusa como matéria prima, portanto não tem relação com o processo de distribuição de gusa (característico de usinas integradas, onde o ferro gusa é obtido através do processo de alto forno), objeto de estudo deste trabalho.

As duas rotas são mostradas na Figura 2, assim como os processos de refino secundário.

Figura 2 – Processo macro de uma aciaria



Estes processos secundários podem ou não ser utilizado no circuito de produção, pois dependem do tipo de aço a ser produzido. Durante o processo primário grande parte das impurezas são oxidadas e removidas, entretanto uma quantidade significativa de oxigênio permanece dissolvido no aço bruto, isso tem impacto negativo nas propriedades mecânicas do aço. A remoção do oxigênio se dá através da dissolução de elementos que possuem maior afinidade com o este do que com o ferro, como por exemplo o Alumínio, Silício e Manganês. Uma característica comum dos processos envolvidos no refino secundário é a injeção de um gás (geralmente argônio) para agitação do aço líquido.

O primeiro processo do refino secundário representado na Figura 2 é o LF – *Ladle Furnace*, também conhecido como forno panela, onde se corrige a composição química do aço através da adição de ferro-ligas e permite o ajuste de temperatura entre estreitas faixas, o que permite a produção de tipos mais complexos de aço. O ajuste de temperatura se dá através dos eletrodos de grafite que fornecem aquecimento para o aço líquido. Além disso, o forno panela também se mostrou ser uma unidade capaz de garantir o equilíbrio térmico entre o aço e o revestimento refratário da panela, o que é considerado um ponto chave para a garantia da qualidade do aço produzido. O segundo processo, denominado RH (*Ruhrstahl Heraeus*), é realizado em dois vasos revestidos com refratário, e dois *snorkels*, imersos no banho de aço, que fazem a comunicação entre estes vasos. Ao reduzir a pressão do sistema, o banho de aço líquido sobe para o vaso de vácuo onde ocorre a descarbonetação e outras reações de desgaseificação.

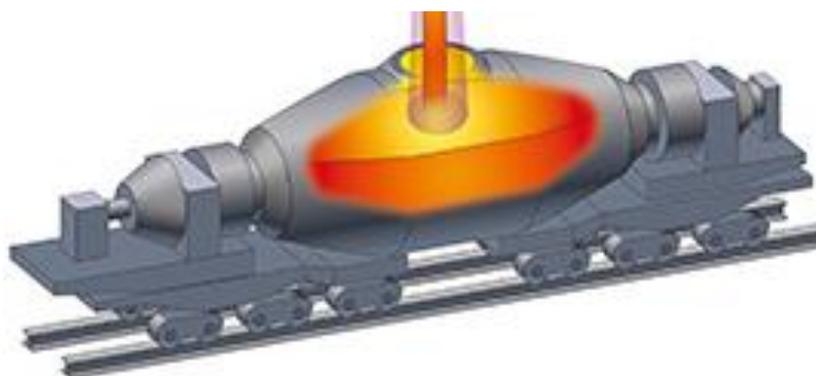
Por fim, o aço acabado é levado ao lingotamento contínuo através de uma panela. Este aço é então transferido à um distribuidor, responsável por alimentar um ou mais moldes, onde o aço é solidificado transformando-se em placas, tarugos ou outros tipos de estruturas. Diversas panelas podem alimentar o distribuidor, que funciona como um *buffer*, o que permite que o fluxo de aço líquido seja constante pelos moldes.

Um problema comum entre o transporte de gusa, o pré-tratamento e o refino secundário é a perda de energia térmica do metal líquido. Manter a massa fundida em uma panela faz com que a temperatura caia a uma taxa de aproximadamente 0,5°C por minuto. Por isso, é frequentemente necessário compensar o calor perdido antes do lingotamento contínuo, para evitar a solidificação prematura e garantir que o lingotamento seja realizado na temperatura correta (MAZUMDAR e EVANS, 2011).

2.3 Transporte do Ferro-Gusa

O transporte de ferro-gusa do alto forno para a aciaria em uma usina siderúrgica é realizado em sua grande maioria por veículos sobre trilhos: carros torpedo tracionados por uma locomotiva. O carro torpedo possui esse nome pelo seu formato – um corpo central cilíndrico com as extremidades cônicas – que lembra um torpedo. Este corpo é revestido internamente com tijolos refratários que minimizam a troca de calor entre o metal líquido e o ambiente externo. Como pode ser visto na Figura 3, um carro torpedo possui uma abertura no seu topo por onde é carregado o ferro gusa no alto forno. Por essa mesma abertura, o metal líquido é transferido para uma panela de recebimento de gusa na aciaria através do basculamento do carro torpedo. Este basculamento consiste no giro do recipiente de armazenamento que é acionado por um motor, permitindo uma rotação de 360°, importante para basculamento e limpeza do mesmo.

Figura 3 – Carro torpedo



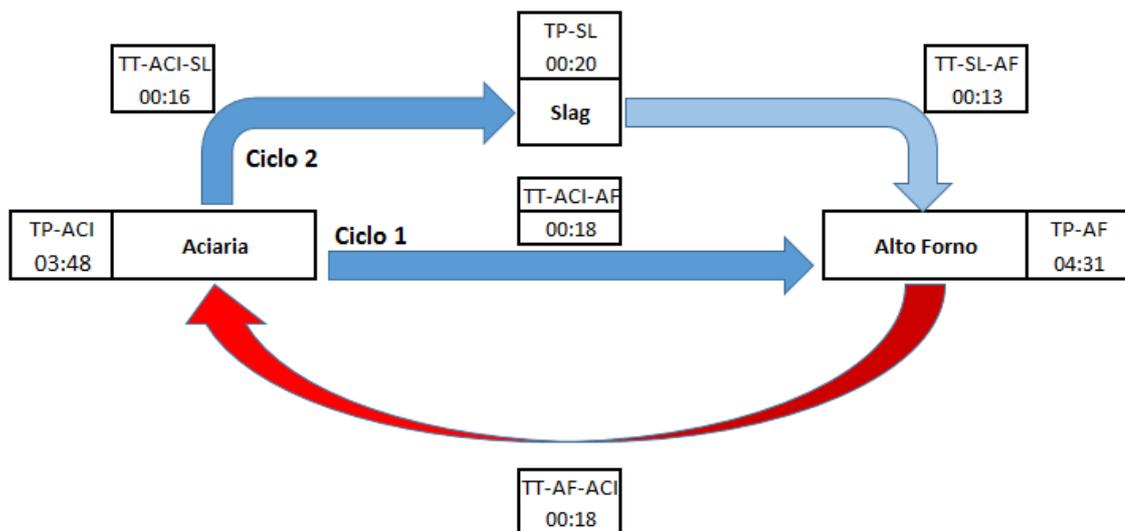
Fonte: Wadeco (2018)

2.3.1 Ciclo do Torpedo e Perdas do Processo

É chamado de ciclo de operação do torpedo cada vez que o mesmo é carregado com ferro-gusa e em seguida descarregado. No ciclo normal de operação, um carro torpedo primeiramente recebe gusa no alto forno; em seguida o mesmo é descarregado na aciaria; e por fim ele retorna ao alto forno para dar início à um novo ciclo. De acordo com uma frequência pré-estabelecida, ou quando a operação detectar que há indícios de “cascão” (acúmulo de gusa ou escória solidificados no interior do torpedo), é incluída uma terceira etapa no ciclo, em que o torpedo passa por uma limpeza na área de descarte de escória, também chamada de SD (*Slag Disposal Pit*), antes de retornar ao alto forno para recebimento de gusa novamente.

A Figura 4 ilustra os dois ciclos principais descritos para a siderúrgica, objeto deste estudo: (1) o torpedão retorna da aciaria para o alto forno; e em (2) o torpedão passa por um processo de limpeza após descarregar na aciaria. Nela está destacado o tempo médio de cada operação. É importante observar que os tempos de espera ou trabalho no alto forno e aciaria são inerentes ao processo, além disso, eles retratam inclusive a espera por uma locomotiva para o início da movimentação. Ambos os ciclos podem ainda ser divididos em duas fases: a primeira representada pela seta vermelha, relaciona-se com o tempo em que o torpedão está carregado com metal líquido e, portanto, tem sua temperatura aumentada graças à transferência de energia térmica do gusa para os tijolos refratários (torpedão). A segunda etapa está representada pelas setas de cor azul e mostra o período em que o torpedão encontra-se vazio, perdendo temperatura para o ambiente.

Figura 4 – Tempo médio para cada etapa dos ciclos de um torpedão

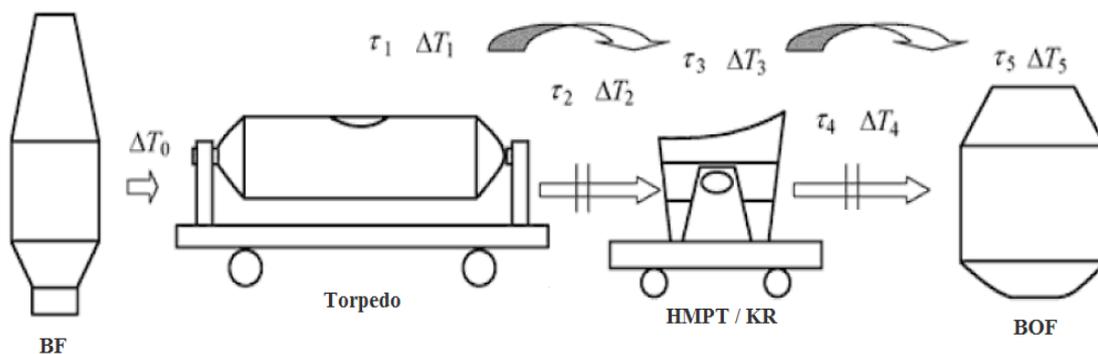


Fonte: Autoria própria (2019)

Entre a produção do ferro-gusa no alto forno e o carregamento deste no convertedor, o metal líquido passa por algumas etapas de troca de recipientes e transporte. Para cada uma destas etapas há uma determinada perda de calor como pode ser visto na Figura 5. A primeira queda de temperatura ΔT_0 ocorre no momento de carregamento do carro torpedão e trata-se da carga térmica perdida pelo gusa para o meio ambiente e para o próprio canal de corrida, ao percorrê-lo. Essa queda de temperatura para o canal é acentuada para o início do vazamento, quando a diferença de temperatura entre o canal e o gusa é maior, assim o canal acaba sendo aquecido pelo fluxo inicial de metal líquido. A segunda queda de temperatura ΔT_1 representa

a temperatura perdida do ferro gusa na transferência de calor para o próprio carro torpedo e é diretamente proporcional ao tempo τ_1 de residência no mesmo. Essa queda relaciona-se com a temperatura interna do torpedo vazio, no momento do enchimento, de forma que, quanto menor a temperatura do torpedo, maior a transferência de calor para o mesmo. Visto isso, é desejável que o carro torpedo a ser utilizado para receber a primeira carga de gusa após a abertura do furo de corrida (início de um vazamento) seja o que possua maior temperatura interna. O que ameniza o impacto da quantidade de calor perdida para o canal na temperatura do gusa, uma vez que se diminui a transferência de calor do gusa para o torpedo. Outro ponto importante é que há um limite de temperatura mínima que o torpedo deve ter para receber uma carga de ferro gusa, pois uma diferença de temperatura muito grande entre o torpedo e o gusa resulta na diminuição do tempo de vida do refratário.

Figura 5 – Etapas de transporte do ferro gusa



Fonte: Adaptado de YIN (2011)

Ainda na Figura 5, é possível observar as variações de temperatura ΔT_2 , ΔT_3 e ΔT_4 , que representam respectivamente: a perda de energia térmica no processo de transferência do gusa do carro torpedo para a panela da aciaria; a perda devido ao processo de pré-tratamento do gusa (desfosforação e dessulfurização); e a perda na transferência para o convertedor.

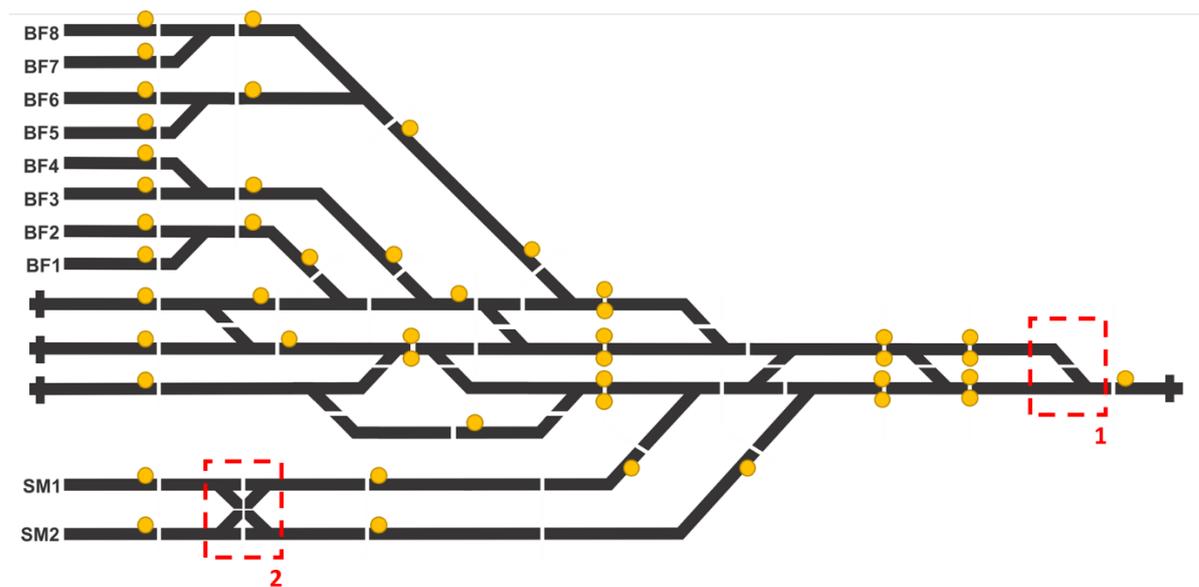
Somente no convertedor o metal líquido passa a receber calor novamente através do processo de oxidação das impurezas, dessa forma ΔT_5 é positiva. Portanto, a redução destas quedas de temperatura é de suma importância para o processo de produção de aço, uma vez que é imprescindível garantir que o metal chegue em estado líquido até o processo do convertedor. Além disso, a temperatura de carregamento do ferro gusa no convertedor dita a quantidade de sucata metálica que pode ser adicionada à corrida. Esta sucata possui um custo muito inferior ao ferro gusa e traz diversos benefícios ao processo por se tratar de um aço praticamente sem impurezas. Além disso, a redução da variação de temperatura do carro torpedo está diretamente

relacionada às quedas de temperatura ΔT_0 e ΔT_1 . Essa redução da variação é importante pois aumenta a vida útil do refratário do torpedo, que acarreta em alguns benefícios como: maior disponibilidade do veículo, redução de custos com manutenção e troca de refratários, e redução de impactos ambientais com o descarte prematuro do material refratário.

2.3.2 A Malha Ferroviária e a Dinâmica Operacional

A malha ferroviária de uma usina siderúrgica tem como função principal prover o transporte do ferro gusa. Ela é composta por diversos componentes. A Figura 6 mostra uma malha simplificada de uma usina siderúrgica considerando uma planta de alto forno com quatro furos de gusa e uma planta de aciaria com dois poços de recebimento. Os pontos destacados com a tag BF representam as linhas de recebimento de gusa sob os furos do alto forno. Os pontos com a tag SM são as linhas de descarga de ferro gusa da aciaria.

Figura 6 – Malha ferroviária simplificada de uma usina siderúrgica



Fonte: Autoria própria (2019).

A malha ferroviária é dividida em segmentos que recebem o nome de trecho e são representados pelos retângulos pretos. Os polígonos em formato de “Y” ou em forma de tesoura (detalhes 1 e 2 destacados na Figura 19, respectivamente) representam as chaves de mudança de via e são responsáveis por permitir que um veículo se desloque de um trecho para outro paralelo. Cada trecho possui sensores de linha que detectam a presença de um veículo quando ocupado. A sinalização da malha é feita através de semáforos, representados pelos círculos

amarelos. Estes têm a função de indicar ao maquinista se determinado trecho está liberado ou não para que a locomotiva possa seguir.

Em um ambiente automatizado, pode-se ter tanto os comandos dos semáforos e das máquinas de mudança de via quanto os sinais de ocupação de trechos ligados a um equipamento de intertravamento eletrônico (EIE – *Electronic Interlocking Equipment*), um tipo de Controlador Lógico Programável (CLP) destinado ao controle e gerenciamento da malha ferroviária. No EIE são programadas todas as lógicas de intertravamento de segurança para o controle de tráfego, onde, por exemplo, um semáforo só dará condição para uma locomotiva seguir para o trecho seguinte caso ele esteja desocupado.

Fazendo a interface com o operador, existe uma HMI (*Human Machine Interface*), ou seja, um sistema de supervisão em que o operador consegue acompanhar a movimentação e localização dos veículos sobre a ferrovia, além de comandar os equipamentos de sinalização. Devido ao ambiente agressivo (altas temperaturas) e ao alto custo de equipamentos para a identificação das locomotivas e torpedos de forma automática, como sensores RFID (*Radio-Frequency IDentification*) por exemplo, na grande maioria dos sistemas encontrados atualmente, os sensores de trecho geram apenas a informação de ocupação ou não do segmento. Portanto uma lógica de rastreamento e consolidação de dados precisa ser criada separadamente. Em alguns sistemas são utilizados um servidor e um sistema de banco de dados para este processamento.

Considerando uma usina siderúrgica dotada de um sistema de rastreamento de veículos e sinalização automatizada, pode-se descrever o processo de transporte e distribuição de gusa da seguinte forma:

1. O sistema do alto forno ou o da aciaria envia o planejamento da próxima corrida para o sistema de supervisão da equipe de distribuição de gusa;
2. O operador da sala de controle de distribuição identifica a quantidade e quais carros torpedo devem ser levados ao alto forno e/ou aciaria para atendimento da demanda;
3. O operador então localiza uma ou mais locomotivas na via para realizar a movimentação do carro torpedo e envia a ordem de movimentação ao maquinista;
4. O operador define o melhor trajeto do ponto de origem ao ponto de destino e realiza a abertura dos semáforos e operação das chaves de mudança de via para dar condição aos veículos de se locomoverem.

A partir do recebimento da requisição de movimentação (item 1) o operador precisa selecionar quais os carros torpedo devem ser levados ao alto forno ou aciaria. Essa decisão

descrita no passo 2 é um tanto complexa pois depende de diversas variáveis e deve levar em consideração se o carro torpedo destina-se ao alto forno ou à aciaria.

Uma das estratégias adotadas para a tomada de decisão de escolha do torpedo para o atendimento ao alto forno relaciona-se à perda de calor para o canal de corrida que, como mencionado na seção anterior, é desejável que o carro torpedo ao receber a primeira carga de gusa de uma corrida tenha a maior temperatura possível. Após a carga do primeiro torpedo, tem-se uma redução considerável desta perda, pois o canal de gusa se encontra em uma temperatura próxima a do metal líquido. Portanto, é desejável que os próximos torpedos sejam priorizados pela menor temperatura, mas que tenham condição de chegar com uma temperatura aceitável na aciaria levando em conta os tempos estimados no ciclo do torpedo da Figura 4. Isso é importante pois a temperatura mínima para carregamento do torpedo deve ser respeitada para prevenir danos ao material refratário. Caso a temperatura do torpedo esteja abaixo do limite mínimo, este deverá ser direcionado para aquecimento em um aquecedor, o que gera gastos relacionados à indisponibilidade do veículo e à energia gasta para aquecimento. Além disso, uma vez carregado, o gusa deve ser entregue à aciaria respeitando uma margem mínima de temperatura para tratamento. Caso esse limite não seja atendido, o gusa deverá ser basculado no pátio de emergência, de forma que o ferro líquido é transformado em sucata metálica. Essa sucata poderá ser carregada posteriormente no convertedor, porém tem um preço muito mais alto que a sucata convencional.

Para a aciaria, quanto maior a temperatura de entrega de gusa, melhor. Entretanto, quando o saldo de gusa é alto (situações em que há vários torpedos já carregados), priorizar a entrega pela temperatura mais alta irá ocasionar perda acentuada de calor de alguns torpedos e, conseqüentemente, este ferro gusa será transformado em sucata. Dessa forma, uma estratégia também adotada é a priorização dos carros com temperatura mais baixa, a fim de diminuir a chance de perda de torpedo por temperatura baixa.

A distância e a localização dos veículos na malha ferroviária devem ser levadas em conta para ambos os casos. Outro fator que também influencia diretamente na qualidade da entrega de ferro gusa é a quantidade de torpedos ativos no ciclo. Sabe-se que quanto menor a quantidade de torpedos no ciclo, maior a taxa de utilização deles, portanto menos perda de energia térmica ocorre, pois eles passam mais tempo carregados e menos tempo vazios. Entretanto quanto menor a quantidade de veículos no ciclo, mais eficiente deve ser o gerenciamento dos torpedos e locomotivas, pois qualquer falha operacional, ou em um dos processos (alto forno ou aciaria), causa impacto quase imediato no outro.

2.4 Importância do Processo de Distribuição de Gusa

Por se tratar de um processo térmico e de resposta lenta, a estabilidade do processo de produção de gusa é essencial para garantir alta produtividade. Para garantir esta estabilidade é essencial o controle de nível do metal líquido no cadinho do alto forno, por isso é de suma importância o atendimento das requisições de carros torpedo conforme programação de corridas demandadas pelas condições operacionais do alto forno. Já durante a produção do aço na aciaria, observa-se uma queda de temperatura muito elevada durante as várias etapas do processo produtivo. Como a temperatura é um parâmetro essencial para o correto lingotamento do aço, essa queda de temperatura deve ser compensada de alguma forma, seja através da redução da quantidade de material adicionado ou através do aumento de tempo de tratamento nos processos de aquecimento, como o forno panela por exemplo, que impactam diretamente nos custos de produção do aço. Portanto, além de ser importante para a estabilidade do processo de alto forno, a otimização do processo de distribuição de gusa, com a redução das perdas de energia térmica neste, é essencial para a redução dos custos de produção do aço. A otimização deste processo está diretamente ligada à otimização de rotas, que será abordada no capítulo seguinte.

3 PROBLEMA DE ROTEAMENTO DE VEÍCULOS

O problema de roteamento de veículos (VRP) foi proposto pela primeira vez por Dantzig e Ramser (1959) e aplicado ao planejamento de rotas para uma frota de caminhões de mesma capacidade na entrega de combustível à diversos postos de gasolina, minimizando a distância total percorrida. Clarke e Wright (1964) apresentaram um modelo linear para o problema de como realizar entregas à um conjunto de consumidores geograficamente espaçados, a partir de um depósito central utilizando uma frota de veículos com capacidades variadas. Em resumo, o VRP consiste em encontrar o conjunto ótimo de rotas para uma frota de veículos de modo a atender as demandas de um grupo de clientes.

Ao longo do tempo, pesquisadores vem estudando e formulando diversas variações para este problema através da incorporação de características de problemas do mundo real. Uma das variações mais populares é o problema de roteamento com janela de tempo (VRPTW – *Vehicle Routing Problem with Time Windows*), em que as visitas aos consumidores devem ser realizadas dentro de um certo intervalo de tempo (TOH; VIGO, 2002b), como por exemplo um serviço de entregas que só realiza determinadas entregas no período da tarde; ou considerando a diferença nos horários de atendimento de determinado consumidor que funciona apenas em horário comercial frente à um consumidor que recebe mercadorias no regime de trabalho de 24hs por dia. Bräysy e Gendreau (2005) e Gendreau e Tarantilis (2010) trazem aplicações do VRPTW.

Outra aplicação muito encontrada na literatura é o problema de roteamento de veículos com coleta e entrega (PDVRP – *Pickup and Delivery Vehicle Routing Problem*) em que o veículo precisa se deslocar da garagem (ou depósito) a um ponto de coleta de material, onde o mesmo será carregado, e posteriormente o veículo segue para um ponto de entrega, onde o material é descarregado. Neste problema, uma mesma rota pode ser composta de um ou mais pontos de coleta e entrega, e ao fim da jornada o veículo deve retornar ao depósito para aguardar uma nova rota (BERBEGLIA *et al.*, 2007). O problema de roteamento de veículos com múltiplas garagens (MDVRP – *Multi-depot Vehicle Routing Problem*) é frequentemente associado ao PDVRP, como em Irnich (2000) e Sombuntham e Kachitvichyanukul (2010). Este problema é caracterizado por possuir mais de uma garagem de veículos, pontos onde estes iniciam e finalizam suas rotas. Uma variação do PDVRP com foco no transporte coletivo de passageiros é o problema de roteamento conhecido como *Dial-A-Ride Problem* (DARP) em que os passageiros solicitam o transporte através de uma requisição. Este problema pode

apresentar diversas restrições relacionadas ao tipo de veículo, janela de tempo e capacidade (MOLENBRUCH; BRAEKERS; CARIS, 2017).

No problema dinâmico de roteamento de veículos (DVRP – *Dynamic Vehicle Routing Problem*) algumas informações necessárias para o planejamento das rotas são conhecidas somente após o início da execução da rota original, assim a natureza estática da rota se torna dinâmica, o que dá origem a uma situação de replanejamento parcial ou total das rotas dos veículos assim que novas requisições são recebidas (ADEWUMI; ADELEKE, 2018).

Um caso particular do problema de roteamento de veículos é o Problema do Caixeiro Viajante (TSP - *Travelling Salesman Problem*). Este problema teve destaque a partir de um jogo criado por Willian Rowan Hamilton em 1857, e consiste em, dado um grafo com uma quantidade n de cidades, encontrar um ciclo, otimizando a distância a ser percorrida, para que todas as n cidades sejam visitadas uma única vez, retornando à cidade de origem (GOLDBARG; LUNA, 2005). A este trajeto foi dado o nome de ciclo hamiltoniano em homenagem ao criador do jogo.

Devido a sua natureza combinatória, o problema de roteamento de veículos é considerado um problema NP-difícil (TOTH; VIGO, 2002a), o que reflete em maior dificuldade de resolvê-lo em um tempo finito aceitável através de algoritmos exatos como programação linear, *branch and bouch* ou o algoritmo de planos de corte. Entretanto, modernas heurísticas, como Otimização por Colônia de Formigas – ACO (YU; YANG; YAO, 2009; BRITO *et al.*, 2015), Algoritmos Genéticos (CHENG; YU, 2013; ABDALLAH; ESSAM; SARKER, 2017) e Otimização por Enxame de Partículas – PSO (OKULEWICZ; MANDZIUK, 2013), têm a habilidade de produzir soluções de alta qualidade em um curto espaço de tempo. Embora estes algoritmos não garantam a solução ótima do problema, sua utilização tem-se mostrado muito eficiente. Na literatura, é amplamente encontrada a aplicação da metaheurística ACO para a resolução dos mais diversos problemas de roteamento de veículos, e diversas variações criadas do algoritmo tem mostrado vantagens através de um grande número de experimentos (XU; PU; DUAN, 2018).

Dhawan e Nassa (2014) apresentam uma revisão bibliográfica dos principais trabalhos com a utilização de ACO para a resolução do VRP, em que se destaca a criação do algoritmo *Ant System* e o seu aprimoramento para a resolução do VRP em meados de 1999 (BULLNHEIMER *et al.*, 1997, 1999). Também em 1999, Su (1999) apresentou um sistema de controle e agendamento dinâmico para roteamento de veículos em múltiplas garagens.

Stodola e Mazal (2016) propõem um algoritmo ACO para o MDVRP onde uma colônia de formigas é empregada para cada depósito. Além disso, são propostos e utilizados

diferentes métodos de seleção de depósitos considerando o percentual de uso, concentração de feromônios entre o depósito e os clientes, ou um fator aleatório.

Nos últimos anos, tem-se notado o surgimento de muitos algoritmos híbridos. Estes algoritmos são caracterizados pela associação de diferentes métodos para a formação de um único algoritmo, com o intuito de compensar as fraquezas ou unir pontos fortes dos algoritmos base utilizados. Euchi, Yassine e Chabchoub (2015) propõem um algoritmo híbrido para solução do problema de roteamento dinâmico de veículos com coleta e entrega (DPDVRP – *Dynamic Pickup and Delivery Vehicle Routing Problem*) associando um algoritmo de busca local 2-opt ao sistema de colônia de formigas. Baseado na ideia de Kilby, Prosser e Shaw (1998), o tempo de trabalho é dividido em intervalos de tempo de tal forma que o problema dinâmico possa ser tratado como um problema parcialmente estático. A ideia de um algoritmo híbrido é mantida por Xu, Pu e Duan (2018) para o DMDVRP adicionando ao ACO, além do algoritmo de busca local, a criação de regiões para cada depósito e um operador de mutação.

Huang e Yu (2017) propuseram novos algoritmos ACO híbridos para resolver o problema de escalonamento de trabalho multi-objetivo, um algoritmo PSO foi usado para o ajuste de parâmetros. Em (TWINKLE *et al.*, 2018) os autores resolvem um DARP com dois objetivos (minimizar o tamanho da frota e minimizar o custo das rotas) com a aplicação de sistemas de colônia de formigas (ACS). É utilizada uma colônia distinta para cada objetivo, e a melhor solução é atualizada com base nos dois sistemas. Zhanga e Xionga (2018) propõem um algoritmo imune ACO para resolver o problema de distribuição de suprimentos em desastres, otimizando uma função multi-objetivo que considera a maximização da satisfação dos pontos de demanda, a minimização dos custos de transporte e serviço, e a minimização do tempo de processamento e entrega.

Uma revisão bibliográfica categorizada sobre a aplicação de metaheurísticas para diferentes problemas de roteamento de veículos é feita em Gendreau *et al.* (2008) e, por fim, uma revisão classificando artigos publicados entre 2009 e 2015 sobre VRP é apresentada em Braekers, Ramaekers e Nieuwenhuyse (2016).

Uma lacuna existente na literatura atual deve-se ao fato de os algoritmos e trabalhos voltados para o problema de roteamento de veículos considerar apenas a distância vetorial como a distância entre os pontos de coleta ou entrega. Dessa forma é considerado apenas o tempo de resolução do problema de roteamento de veículos, desprezando-se o tempo necessário para calcular a distância entre dois pontos da rota. Em sistemas do mundo real, essa distância geralmente não é conhecida e, devido aos milhares de conjuntos possíveis de pontos de origem e destino para a malha ferroviária, bem como a alteração das condições e restrições de uso de

determinado trecho, essa distância deve ser calculada em tempo real, o que onera tanto o tempo quanto o nível de processamento. Este trabalho objetiva preencher esta lacuna através da aplicação do algoritmo de colônia de formigas para encontrar o menor caminho entre estes pontos, de forma a possibilitar a utilização do algoritmo de otimização em sistemas de controle.

A Seção 3.1 apresenta o modelo matemático para o problema de roteamento de veículos aplicado à malha ferroviária de uma usina siderúrgica integrada. Já a Seção 3.2 mostra os princípios da otimização por Colônia de Formigas. O capítulo encerra com a Seção 3.3 que integra o problema de roteamento à otimização por colônias de formigas, introduzindo a proposta de otimização abordada no Capítulo 4.

3.1 Modelo Matemático para o Problema de Roteamento

O problema foi modelado com base no modelo proposto em (KACHITVICHYANUKUL *et al*, 2015) para o problema generalizado de roteamento de veículos com múltiplos depósitos e múltiplas requisições de coleta e entrega (GVRP-MDMPDR – *Generalized Vehicle Routing Problem for Multiple Depots with Multiple Pickup and Delivery Requests*). As mudanças básicas neste modelo podem ser resumidas na inclusão das restrições de temperatura, e inclusão da componente temperatura na função objetivo. Dessa forma, a função objetivo considera a distância total percorrida, a quantidade de veículos utilizada e a penalização por valores de temperatura estimados.

Por fim, as seguintes premissas são adotadas para o modelo: (1) os pontos de coleta representam a localização dos carros torpedo aguardando uma movimentação; (2) pontos de entrega são nós de destino de cada torpedo; (3) os depósitos estão relacionados diretamente com a localização de cada locomotiva, portanto a restrição de que cada veículo deve retornar ao seu depósito foi removida da formalização do problema; (4) a capacidade da locomotiva atual se limita a dois carros torpedo, entretanto outras configurações podem ser adotadas a depender do tipo de veículos utilizados.

3.1.1 Conjuntos

Abaixo são descritos os conjuntos utilizados no modelo:

- P :** Conjunto de pontos de coleta $P = \{1, 2, \dots, R\}$, onde R é o número total de requisições; representa a localização do carro torpedo aguardando uma movimentação;
- D :** Conjunto de pontos de entrega $D = \{1 + R, 2 + R, \dots, 2R\}$, onde $r + R$ representa o nó de destino da requisição r ; e o nó de destino do carro torpedo;
- N :** Conjunto de todos os pontos de *pickup* e *delivery* $N = P \cup D$;
- N_k** Subconjunto dos nós visitados pelo veículo k ;
- K :** Conjunto de veículos;
- G :** Conjunto de depósitos, onde τ_k representa o ponto de origem do veículo k e τ'_k representa o ponto final da requisição;
- V :** Conjunto de todos os nós $V = N \cup G$;
- A :** Conjunto dos arcos (i, j) que representa um trecho ligando o nó i ao nó j , onde $i, j \in V$;

3.1.2 Parâmetros

Os parâmetros de entrada do modelo são:

- C_k :** Capacidade do veículo k ;
- f_k :** Custo fixo do veículo k , caso ele seja utilizado;
- d_{ij} :** Distância entre os nós i e j ;
- t_{ij} :** Tempo estimado entre os nós i e j ;
- t_{sp}^i :** Tempo fixo de serviço para operações de coleta no ponto i ;
- t_{sd}^i :** Tempo fixo de serviço para operações de delivery no ponto i ;
- $[a_i, b_i]$:** Janela de tempo para atendimento da requisição, em que uma coleta ou entrega no ponto i só pode ser realizada entre os tempos a_i e b_i ;

- l_i : Quantidade de carros torpedo a serem carregados no nó $i, i \in P$;
- u_j : Quantidade de carros torpedo a serem descarregados no nó $j, j \in D$;
- T_d^{jk} : Temperatura estimada do carro torpedo no momento da entrega pelo veículo k no ponto $j, j \in D$;
- T_0^{jk} : Temperatura no momento da última movimentação do carro torpedo (quando recebeu ou descarregou gusa) coletado pelo veículo k para atendimento da demanda no ponto $j, j \in D$;
- T_{min}^j : Temperatura mínima de entrega do torpedo no ponto $j, j \in D$;

3.1.3 Variáveis de Decisão

A seguir são descritas as variáveis de decisão utilizadas no modelo:

- x_{ijk} : Variável binária igual a 1 se o veículo k viaja do nó i para o nó j , e de valor 0, caso contrário;
- t_s^{ik} : Tempo em que o veículo k iniciou o serviço em i ;
- L_{ik} : Quantidade total de torpedos carregados pela locomotiva k ao passar pelo ponto i , com $i \in V$ e $k \in K$;

3.1.4 Modelo do Problema

O modelo matemático do problema é detalhado como um problema de programação linear inteira conforme definido em (KACHITVICHYANUKUL *et al*, 2015), e adaptado de modo a incorporar as variáveis de temperatura presente no processo de distribuição de gusa. Dessa forma tem-se como objetivo minimizar a soma ponderada da distância total percorrida, do custo fixo por veículo e do custo relacionado à variação de temperatura:

$$\min \alpha \sum_{k \in K} \sum_{(i,j) \in A} d_{ij} x_{ijk} + \beta \sum_{k \in K} \sum_{j \in P} f_k x_{\tau_k, j, k} + \gamma \sum_{k \in K} \sum_{j \in D} |T_d^{jk} - T_0^{jk}| \quad (3.1)$$

Sujeito a:

$$\sum_{j \in V} x_{ijk} - \sum_{j \in V} x_{i,j+r,k} = 0 \quad \forall k \in K, \forall i \in P \quad (3.2)$$

$$\sum_{j \in P} x_{\tau_k,j,k} = 1 \quad \forall k \in K \quad (3.3)$$

$$\sum_{i \in D} x_{i,\tau'_k,k} = 1 \quad \forall k \in K \quad (3.4)$$

$$\sum_{i \in V} x_{ijk} - \sum_{i \in V} x_{jik} = 0 \quad \forall k \in K, \forall j \in N \quad (3.5)$$

$$x_{ijk} = 1 \Rightarrow t_s^{ik} + t_{sp}^i * l_i + t_{sd}^i * u_i + t_{ij} \leq t_s^{jk} \quad \forall k \in K, \forall (i,j) \in A \quad (3.6)$$

$$a_i \leq t_s^{ik} \leq b_i \quad \forall k \in K, \forall i \in V \quad (3.7)$$

$$t_s^{ik} \leq t_s^{i+r,k} \quad \forall k \in K, \forall i \in P \quad (3.8)$$

$$x_{ijk} = 1 \Rightarrow L_{ik} - u_i + l_i = L_{jk} \quad \forall k \in K, \forall (i,j) \in A \quad (3.9)$$

$$L_{ik} \leq C_k \quad \forall k \in K, \forall i \in V \quad (3.10)$$

$$L_{\tau_k k} = L_{\tau'_k k} = 0 \quad \forall k \in K \quad (3.11)$$

$$T_d^{jk} \geq T_{min}^j \quad \forall j \in D \quad (3.12)$$

$$x_{ijk} \in \{0,1\} \quad \forall k \in K, \forall (i,j) \in A \quad (3.13)$$

$$S_{ik} \geq 0 \quad \forall k \in K, \forall i \in V \quad (3.14)$$

$$L_{ik} \geq 0 \quad \forall k \in K, \forall i \in V \quad (3.15)$$

O custo relacionado à temperatura é calculado como a variação da temperatura entre a última operação do torpedo e a previsão de utilização do mesmo, contida na requisição de movimentação. Por exemplo, caso o torpedo esteja vazio e seja selecionado para atender à requisição de um torpedo para o alto forno, a variação de temperatura é calculada como a temperatura no momento da descarga do torpedo na aciaria menos a temperatura estimada no momento em que o torpedo for receber ferro-gusa no alto forno. Dessa forma busca-se minimizar a variação da temperatura do torpedo, que caso vazio irá garantir maior vida útil do material refratário além de maior temperatura do mesmo no momento de recebimento de torpedo. E para o caso do torpedo cheio, esta variável irá garantir a menor perda de calor possível para o ambiente.

A Equação 3.2 determina que caso um veículo visite um ponto de coleta, ele também visitará um ponto de entrega, em que x_{ijk} representa a coleta, e $x_{i,j+r,k}$ a entrega realizada pelo veículo k . A Equação 3.3 determina que uma locomotiva parta de seu ponto de origem (depósito) representado pelo ponto τ_k , e a Equação 3.4 que uma locomotiva encerre seu trajeto no ponto de destino da requisição representado por τ'_k . A Equação 3.5, associada às duas anteriores, define que uma rota é formada por trajetos consecutivos entre o ponto inicial τ_k (localização da locomotiva) e o ponto de última entrega τ'_k para cada locomotiva. A Equação 3.6 define que, para um trajeto entre os pontos i e j , o tempo total gasto com deslocamento t_{ij} e com as operações de coleta t_{sp}^i e entrega t_{sd}^i deve ser inferior ao tempo de início do serviço em j . Já a Equação 3.7 garante que o tempo de serviço t_s^{ik} realizado em i respeita as restrições de tempo definidas por $[a_i, b_i]$. A Equação 3.8 garante que a coleta seja efetuada antes da entrega. As Equações 3.9 a 3.11 tratam das limitações de carga dos veículos, de forma que a primeira equação define que a carga no ponto j é igual à diferença entre a quantidade de carros torpedo coletados (l_i) e entregues (u_i) em i . A Equação 3.10 mostra que a quantidade de torpedos tracionados pela locomotiva deve se limitar à capacidade C_k , e a Equação 3.11 que todo torpedo anexo a locomotiva k deve ser coletado e entregue entre os pontos τ_k e τ'_k . A Equação 3.12 representa a restrição de temperatura mínima T_{min}^j para entrega de torpedos. E, por fim, as Equações 3.13 a 3.15 são restrições gerais que garantem o correto dimensionamento de algumas das variáveis.

3.2 Otimização por Colônia de Formigas

O problema de roteamento de veículos (VRP) tem sido foco de estudos de diversos pesquisadores ao longo dos anos, e diferentes algoritmos foram propostos e aplicadas para a resolução deste problema. Por ser um problema classificado com NP-difícil, as metaheurísticas tem um papel de destaque entre estes algoritmos, uma vez que apresentam uma solução, em geral, muito próxima da solução ótima com um tempo consideravelmente menor que os algoritmos exatos. Entre estes algoritmos, a otimização por colônia de formigas tem o seu destaque devido à eficiência, facilidade de implementação e também à possibilidade de ser executado continuamente, se adaptando às mudanças em tempo real.

Como o próprio nome sugere, a otimização por colônia de formigas baseia-se no comportamento das formigas em busca de alimentos. Este comportamento, bem como sua sociabilidade, divisão de trabalho e organização em forma de colônias, tem intrigado o ser

humano há algumas décadas, existindo relatos de estudos a cerca deste comportamento datados de 1927. Destes estudos, e das observações realizadas em sequência, Marco Dorigo (1992) propôs o primeiro algoritmo baseado no comportamento exploratório das formigas, chamado de *Ant System* (AS). Este trabalho deu início a uma série de outras pesquisas que resultaram na criação de diversos algoritmos também inspirados no comportamento forrageiro das formigas.

Embora muitas versões de algoritmos ACO tenham surgido, suas principais características estão, de forma geral, relacionadas a uma combinação dos primeiros algoritmos: Sistema de Formigas (*Ant System* – AS) (DORIGO, 1992), Sistema de Formigas Elitista (ASelite) (DORIGO *et al.*, 1996), Sistema de Formigas Baseado em Rank (ASrank) (BULLNHEIMER *et al.*, 1997), Sistema de Colônia de Formigas (*Ant Colony System* – ACS) (DORIGO; GAMBARELLA, 1997) e o Sistema de Formigas MAX-MIN (MMAS) (STUTZLE; HOOS, 2000). Além disso, uma tendência atual é a criação de algoritmos híbridos, combinando determinadas características do algoritmo de otimização por colônias de formigas com outros algoritmos de otimização ou metaheurísticas, como por exemplo nos trabalhos de Xu, Pu e Duan (2018), que apresentou um algoritmo de busca local e um operador de mutação adicionados ao algoritmo ACO, e em Ferreira e Braga (2018) em que um controlador *Fuzzy* é proposto para otimização dos ajustes de um sistema de colônias de formigas em tempo real.

Destes algoritmos, o ACS tem destaque pela busca paralela no espaço de soluções e pela eficiência sobre outros modelos como destacado pelos autores (ASMAR, D.; ELSHAMLI, A.; AREIBI, S., 2005). Além disso, a possibilidade de ser executado continuamente contribuiu para a escolha do ACS como algoritmo para otimização do problema descrito na seção anterior.

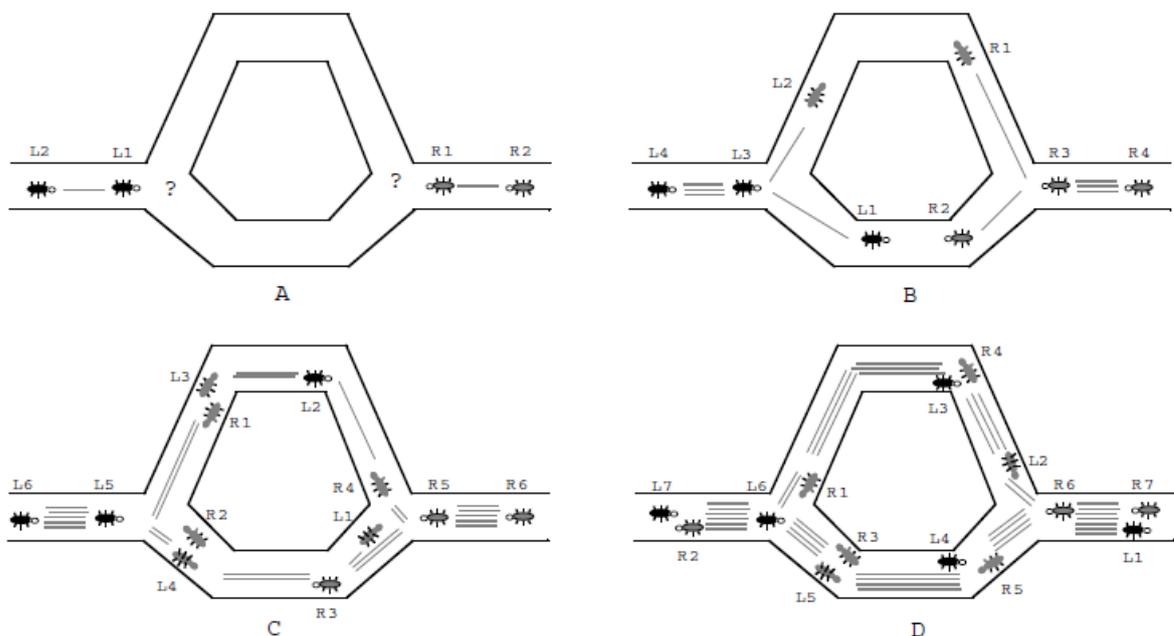
3.2.1 *Ant System* (AS)

Na natureza, uma única formiga é incapaz de se comunicar ou efetivamente buscar comida, mas um grupo de formigas, possui a habilidade de encontrar e coletar comida de forma eficiente para sua colônia. As formigas se comunicam através de uma substância química chamada feromônio. Ao se locomover, cada formiga deposita uma determinada quantidade dessa substância que pode ser seguida por outras formigas. Cada formiga se desloca de forma aleatória, mas quando encontra um rastro de feromônio ela pode optar por segui-lo ou não. Caso a formiga decida por seguir o rastro, ela também deposita feromônio pelo trajeto, aumentando ainda mais sua concentração, e, com o aumento da concentração, também aumenta a

probabilidade de outras formigas escolherem o mesmo percurso. Portanto quanto mais formigas andam por um determinado caminho, maior a atratividade deste percurso para as outras formigas. Além disso, quanto menor um caminho até o alimento, mais rápido uma formiga consegue fazer o trajeto de ida e volta (em busca de comida e o retorno até a colônia), e consequentemente aumenta a concentração de feromônio neste percurso. Com o passar do tempo, a concentração de feromônio diminui gradativamente devido ao processo de evaporação, o que contribui para que caminhos mais longos se tornem menos atrativos.

Um exemplo do comportamento das formigas é representado na Figura 7. No quadro A, as formigas chegam a um ponto de decisão (bifurcação do caminho em dois ramos). Observa-se que neste momento ainda não se tem concentração de feromônios pelo trajeto, uma vez que eles ainda não foram percorridos. O feromônio é representado pelos traços desenhados ao longo dos caminhos. No quadro B, de forma aleatória, algumas formigas escolhem a parte superior e algumas o caminho inferior. No terceiro quadro (C), como as formigas se movem a uma velocidade aproximadamente constante, as formigas que escolhem o caminho de baixo (mais curto) alcançam o ponto de decisão oposto mais rápido do que aquelas que escolhem o caminho superior e mais longo, portanto o feromônio se acumula em uma taxa maior no caminho mais curto, como pode ser observado no quadro D. O número de linhas tracejadas é aproximadamente proporcional à quantidade de feromônio depositada pelas formigas.

Figura 7 – Comportamento das formigas



O *Ant System* (AS) foi aplicado primeiramente ao problema do caixeiro viajante (*Traveling Salesman Problem - TSP*). Dado um grafo $G = (V, A)$ onde V é o conjunto de cidades, e $A = \{(r, s) : r, s \in V\}$ o conjunto de caminhos entre estas cidades com um custo $\delta(r, s)$ associado à distância entre as cidades, o TSP consiste em encontrar o caminho de menor custo que tenha início e fim na mesma cidade e percorra todas cidades do grafo apenas uma vez.

O *Ant System* utiliza uma representação gráfica em que além da medida de custo $\delta(r, s)$, cada borda (r, s) também tem uma medida de atratividade $\tau(r, s)$, chamada feromônio, que é atualizada em tempo de execução por formigas artificiais. Informalmente, o *Ant System* funciona da seguinte maneira: (1) cada formiga gera um tour completo escolhendo as cidades de acordo com uma regra de transição de estado probabilística – as formigas preferem se mover para cidades conectadas por bordas curtas com uma quantidade alta de feromônio, e (2) uma vez que todas as formigas tenham completado seus percursos, uma regra de atualização global de feromônios é aplicada. Nessa regra, uma fração do feromônio evapora em todas as bordas (bordas que não são atualizadas se tornam menos atrativas), e então cada formiga deposita uma quantia de feromônio em bordas que pertencem ao seu percurso na proporção de quão curto foi este percurso (em outras palavras, bordas que pertencem a muitos caminhos curtos são as bordas que recebem a maior quantidade de feromônio). Este processo se repete por um número definido de iterações ou até atingir uma estagnação na taxa de melhora do custo do menor caminho encontrado.

O processo de atualização de feromônios é descrito a seguir. Seja S_k o caminho encontrado pela formiga k , a quantidade depositada de feromônio pela formiga na aresta (i, j) é dada por

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k}, & \text{quando aresta } (i, j) \in S_k \\ 0, & \text{quando aresta } (i, j) \notin S_k \end{cases} \quad (3.16)$$

onde Q é uma constante de atualização de feromônio e L_k é o comprimento da aresta (i, j) . A taxa de atualização de feromônio é descrita pela Equação 3.17:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (3.17)$$

onde ρ é a taxa de evaporação de feromônio e m é o número total de formigas.

Por fim, a probabilidade da formiga k , localizada no vértice i , escolher o próximo vértice j , é dada pela Equação 3.18, onde η representa a qualidade da aresta, α e β são

parâmetros de influência do feromônio e da qualidade da aresta, e N_j^k é o conjunto de pontos possíveis de serem visitados a partir do nó i .

$$p_{ij}^k = \frac{\tau_{ij}^\alpha + \eta_{ij}^\beta}{\sum_{l \in N_j^k} \tau_{il}^\alpha + \eta_{il}^\beta} \quad (3.18)$$

3.2.2 *Ant Conlony System (ACS)*

O sistema de colônias de formigas (ACS) foi proposto por Dorigo e Gambardella (1997) como uma melhoria para o AS. Apesar do princípio de funcionamento deste também retratar o comportamento forrageiro das formigas, pode-se destacar algumas diferenças básicas entre os dois algoritmos. A primeira é que uma regra de transição diferente é utilizada para seleção do próximo caminho. Essa regra de transição fornece um caminho direto para o equilíbrio entre a exploração de novas arestas e a exploração do conhecimento acumulado sobre o problema. A segunda diferença se dá na regra de atualização de feromônios, que é aplicada ao final do percurso de cada formiga. E por fim, a terceira diferença está na criação de uma segunda regra de atualização de feromônios que é utilizada apenas para depósito do feromônio da formiga que realizou o melhor percurso. Para diferenciar as duas regras, a primeira passa a se chamar regra de atualização local, e a última, regra de atualização global de feromônios.

De uma forma geral, o ACS aplicado ao problema do caixeiro viajante funciona da seguinte maneira: uma quantidade m de formigas são inicialmente posicionadas em n cidades escolhidas conforme alguma regra de inicialização, que pode ser uma escolha aleatória, por exemplo. Cada formiga então constrói um percurso, ou seja, uma solução viável para o TSP, aplicando repetidamente uma regra pseudo-aleatória (Eq. 3.19) para a escolha do próximo nó a ser visitado. Ao construir seu percurso, uma formiga também modifica a quantidade de feromônio nas bordas visitadas, aplicando a regra de atualização local de feromônios. Uma vez que todas as formigas tenham terminado seu percurso, a quantidade de feromônio nas arestas é modificada novamente, aplicando-se a regra de atualização global. Como no caso do AS, as formigas são guiadas, na construção de seus passeios, por informações heurísticas em que elas preferem escolher bordas curtas, e por informações de feromônio: uma borda com uma alta quantidade de feromônio se torna mais atrativa do que uma que possui baixa concentração. As regras de atualização de feromônios são projetadas de forma que tendem a dar mais feromônio às arestas que devem ser visitadas pelas formigas.

No ACS, a regra de escolha do próximo caminho a ser percorrido por uma formiga é descrito pela Equação 3.19:

$$j = \begin{cases} \arg \max_{u \in N_i^k} \{\tau_{iu} \eta_{iu}^\beta\} & \text{se } q \leq q_0 \\ J & \text{se } q > q_0 \end{cases} \quad (3.19)$$

onde τ é a concentração de feromônios e η representa a qualidade da aresta; q é um número sorteado randomicamente entre 0 e 1, e $q_0 \in [0,1]$ é um parâmetro pré-definido pelo usuário; $J \in N_i^k$ representa o próximo nó selecionado aleatoriamente com uma probabilidade:

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{u \in N_i^k} \tau_{iu}^\alpha + \eta_{iu}^\beta} \quad (3.20)$$

sendo α parâmetros de influência do feromônio e N_i^k o conjunto de pontos factíveis a serem visitados a partir do nó i .

Esta regra possibilita o favorecimento do nó com maior concentração de feromônios e de menor tamanho, representado por $\tau_{iu} \eta_{iu}^\beta$, contra o método padrão de tomada de ação também presente no AS. Isso permite que seja feito um ajuste entre a exploração de novos caminhos (busca global) e a intensificação das buscas nos caminhos já conhecidos (busca local).

Como mencionado anteriormente, no ACS a atualização de feromônios é dada em dois momentos: após a realização do percurso, cada formiga atualiza os feromônios nas arestas percorridas segundo a equação de atualização local (Equação 3.21); a atualização global de feromônios, descrita na Equação 3.22 é realizada ao final da iteração, onde a formiga que realizou o melhor percurso (bs - *best*) pode reforçar o depósito de feromônios sobre as arestas deste percurso. E o valor perdido por evaporação é subtraído junto com as atualizações global e local.

$$\tau_{ij} = (1 - \rho_2)\tau_{ij} + \rho_2\tau_0 \quad (3.21)$$

$$\tau_{ij} = (1 - \rho_1)\tau_{ij} + \rho_1\Delta\tau_{ij}^{(bs)}, \forall (i, j) \in T_{bs} \quad (3.22)$$

onde ρ_1 e ρ_2 são as taxas de evaporação de feromônios global e local, respectivamente; τ_0 representa um nível mínimo de feromônio; T_{bs} representa a melhor solução encontrada; e

$$\Delta\tau_{ij}^{(bs)} = \begin{cases} \frac{1}{f(x_{bs})} & \text{se } (i, j) \in T_{bs} \\ 0 & \text{caso contrário} \end{cases} \quad (3.23)$$

sendo $f(x_{bs})$ uma função do tamanho da aresta x ou custo associado à mesma.

3.3 Metaheurística de Colônia de Formigas Aplicada ao Roteamento de Veículos

Devido às características do processo de distribuição de gusa, o problema a ser otimizado mais se assemelha ao problema de roteamento de veículos com múltiplas garagens e múltiplas requisições de *pickup* e *delivery*. As múltiplas garagens são os pontos de partida dos veículos de entrega. No problema proposto, estes veículos são as locomotivas, e como não possuem um determinado local de espera após cada movimentação, as garagens passam a ser adotadas como a localização atual da locomotiva, ou seja, o ponto de partida da locomotiva para atendimento de uma nova demanda será o mesmo local da última movimentação.

Outra característica da malha ferroviária de uma siderúrgica é o dinamismo em que as mudanças de configuração ocorrem nas vias. Por exemplo, os componentes das vias demandam manutenções periódicas e, portanto, os trechos tornam-se indisponíveis para utilização por um determinado período de tempo. Além disso, por questões de segurança, em uma ferrovia cada trecho pode ser ocupado por apenas um veículo por vez, desta forma, trechos ocupados por outras locomotivas ou carros torpedo também ficam indisponíveis para utilização. Estas mudanças constantes na configuração da malha ferroviária tornam obrigatório adotar uma forma de otimização e cálculo do melhor trajeto em tempo real, similar ao funcionamento de um sistema de navegação veicular por GPS (*Global Positioning System*). O que adiciona uma dificuldade a mais na otimização frente aos demais problemas de roteamento de veículos encontrados na literatura, uma vez que nestes a distância entre dois pontos, geralmente, é conhecida.

Devido a esta necessidade de adaptação às mudanças das vias em tempo real e à exigência de uma rápida resposta para utilização em sistemas industriais, optou-se pela escolha de um sistema de colônia de formigas como metaheurística base para a otimização do problema proposto na Seção 3.1. O ACS tem destaque pela busca paralela no espaço de soluções e pela possibilidade de ser executado continuamente, o que aumenta a velocidade de convergência para boas soluções. O algoritmo proposto será abordado no Capítulo 4.

4 GERENCIAMENTO POR COLÔNIA DE FORMIGAS DA MALHA FERROVIÁRIA DE UMA USINA SIDERÚRGICA

Além das características do problema proposto relacionadas ao roteamento de veículos, a otimização do processo de distribuição de gusa deve considerar também os requisitos de temperatura do torpedo e do metal líquido para o processo de produção de aço. A necessidade de conhecer a temperatura do carro torpedo é essencial para a tomada de decisão de qual carro torpedo direcionar para cada demanda. Além disso, as restrições de temperatura são requisitos que impactam diretamente a qualidade do produto final ou a vida útil do material refratário do carro torpedo.

Este capítulo aborda o funcionamento do algoritmo proposto para a solução do problema de roteamento da malha ferroviária de uma usina siderúrgica. Na Seção 4.1 o algoritmo proposto é apresentado, e seu funcionamento é detalhado. Na Seção 4.2 são propostos os modelos de estimação da temperatura interna do carro torpedo. Por fim, a Seção 4.3 encerra o capítulo abordando os critérios para verificação do desempenho da solução proposta. O simulador criado para a realização de testes do algoritmo e validação dos resultados será apresentado no Capítulo 5.

4.1 Algoritmo Proposto

As demandas do alto forno e aciaria são expressas através das requisições criadas pelo simulador, que representam as próprias solicitações de carro torpedo por parte do processo. O atendimento dessas demandas de forma eficaz é essencial para a disponibilidade operacional e redução de custos. O algoritmo de otimização proposto é executado paralelamente ao simulador e processa as requisições, associando uma solução para cada uma delas.

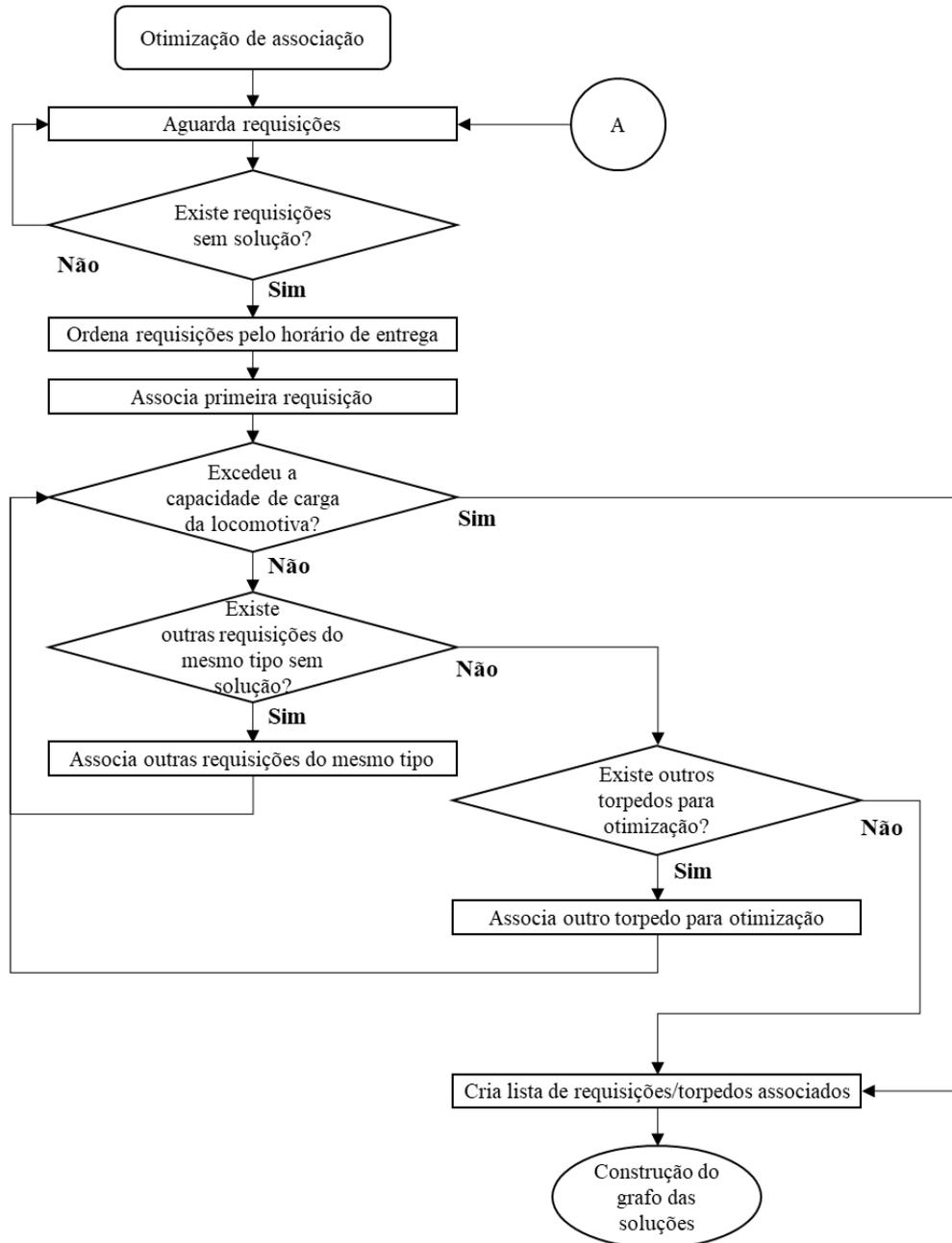
As requisições são divididas em dois tipos: (1) requisições para o alto forno em que o torpedo deve estar vazio para receber gusa, ou (2) requisições para a aciaria em que o carro torpedo deve estar carregado de ferro gusa. O primeiro passo do algoritmo é a tentativa de associação de requisições do mesmo tipo – o que significa que, na grande maioria, teriam pontos de partida e destino próximos. No processo da usina siderúrgica base deste trabalho, a capacidade de tração da locomotiva é limitada em dois carros torpedo.

Caso não haja outra requisição do mesmo tipo para associação, uma segunda otimização é realizada. Nesta buscam-se outros torpedos que estejam disponíveis para a movimentação, por exemplo: um torpedo que já descarregou na aciaria, ou um torpedo já

carregado no alto forno. Estes torpedos são considerados para associação à movimentação original mesmo que não exista uma segunda requisição.

Este processo tem como finalidade otimizar a associação das requisições para cada viagem de locomotiva, e garantir que um número mínimo de locomotivas seja necessário. O fluxograma que descreve este processo de otimização é exibido na Figura 8.

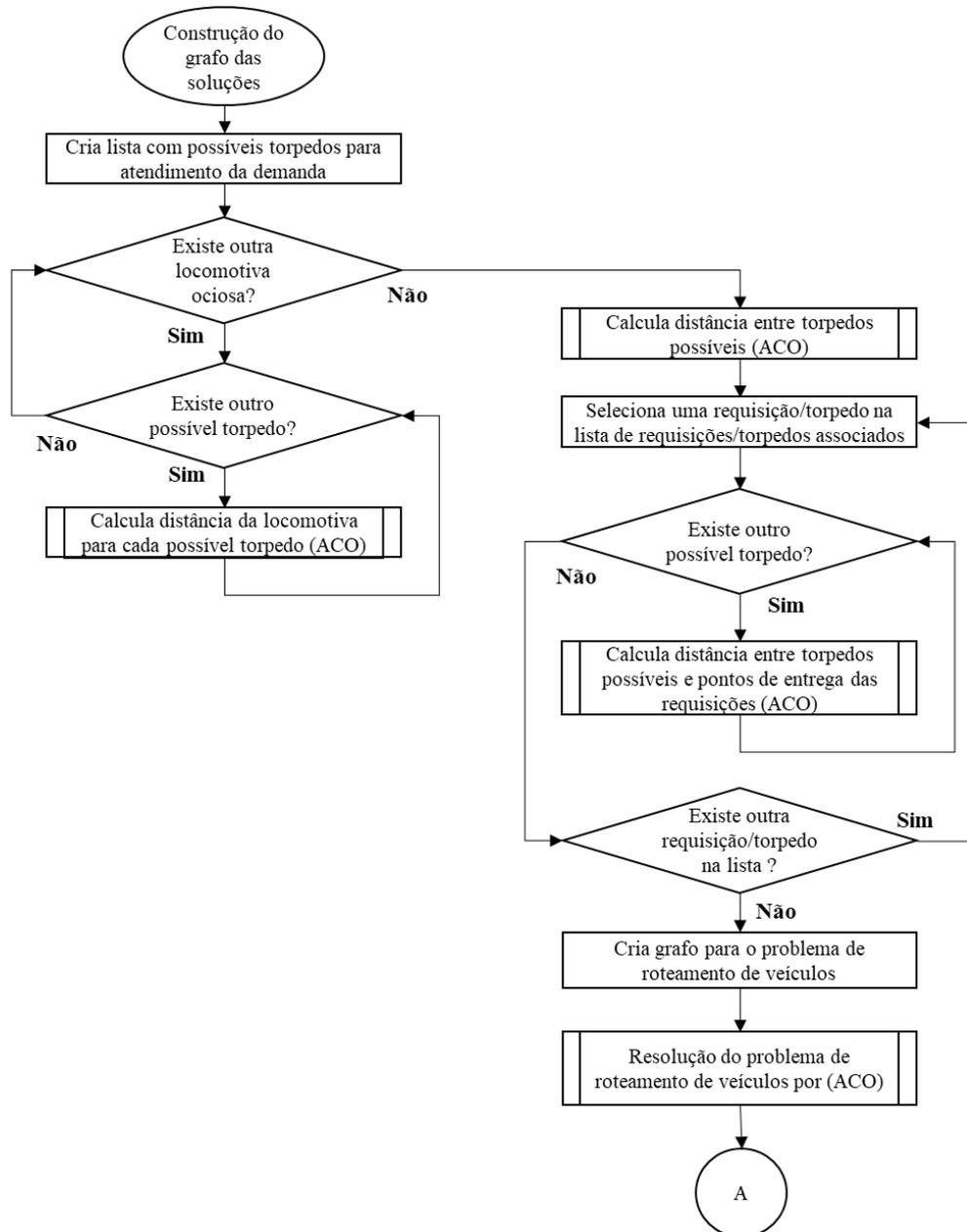
Figura 8 – Fluxograma do processo de otimização de associação das requisições



Fonte: Autoria própria (2019).

Neste processo, uma lista de requisições e torpedos é gerada para iniciar o processo de construção de rotas. Nele, as requisições sem solução são ordenadas pelo horário de atendimento, então a requisição mais próxima é atendida e um conjunto de possíveis associações com outras requisições ou torpedos é gerado para a construção do grafo que representa o problema de roteamento de veículos. O algoritmo para construção do grafo do problema de roteamento de veículos é a sequência do algoritmo de otimização de associação. Este é representado pelo fluxograma da Figura 9.

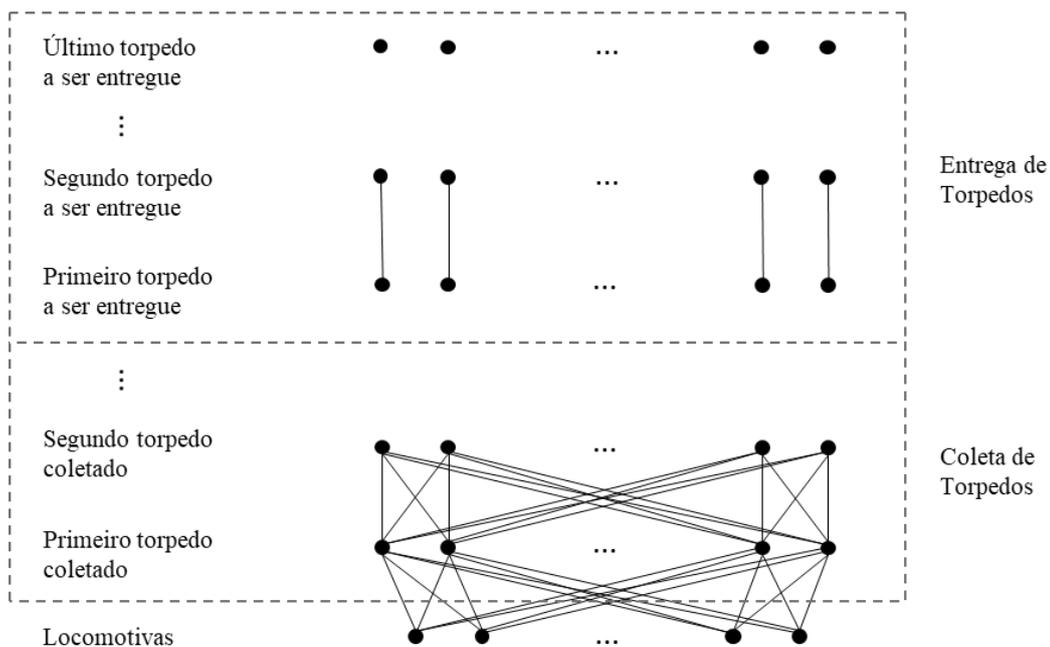
Figura 9 – Fluxograma do algoritmo de construção do grafo das soluções



O primeiro passo para a construção deste grafo é calcular o custo entre os pontos de coleta e entrega de torpedos. Dessa forma, verifica-se inicialmente quais torpedos são compatíveis com a requisição selecionada, e calcula-se a distância entre cada locomotiva ociosa e estes torpedos. Em seguida, a partir da lista de associação gerada pelo algoritmo representado na Figura 8, calcula-se o custo de deslocamento entre os possíveis pontos de coleta de torpedos, sucessivamente até percorrer toda lista de requisições/torpedos associados. O peso associado à cada percurso relaciona-se com a distância e o custo de temperatura. O custo associado à fase de entrega dos torpedos, ou seja, entre os torpedos e os pontos de entrega, é calculado em seguida. Por fim, o grafo para o problema de roteamento de veículos é criado ligando as locomotivas a todos os possíveis pontos de coleta e, em seguida, aos pontos de entrega. Após a criação e resolução do grafo, volta-se ao processo de espera de novas requisições – interconexão entre os dois processos representada por “A” (vide Figura 8).

Neste novo grafo, os vértices representam cada etapa de coleta e entrega de torpedos, e às arestas associa-se o custo referente à temperatura e à distância a ser percorrida. Na Figura 10 é possível ver um exemplo simplificado do grafo para o problema de roteamento de veículos. A solução tem início pela parte inferior com as locomotivas, que por sua vez devem se deslocar à posição de cada torpedo a ser coletado, a depender da combinação de requisições e capacidade da locomotiva.

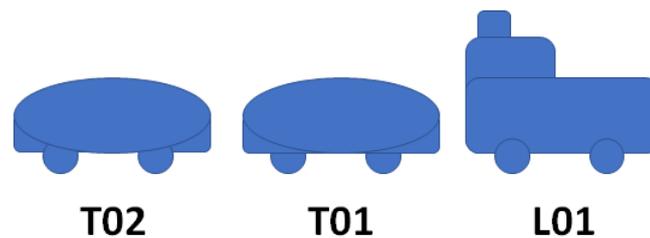
Figura 10 – Grafo criado para a solução final do problema



Fonte: Autoria própria (2019).

Após a coleta de cada torpedo, inicia-se a fase de entregas para o conjunto locomotiva-torpedos, até que todos os torpedos sejam entregues. Observe que na etapa de entrega há uma redução significativa da complexidade do grafo. Isso ocorre porque os torpedos devem ser entregues de acordo com a ordem em que foram coletados. A movimentação de carros torpedo deve obedecer às regras do sistema LIFO (*Last In, First Out*), ou seja, o último carro a ser acoplado à locomotiva deve ser o primeiro a ser entregue. Com base na Figura 11 é possível ver que isso deve-se à disposição em fila dos carros torpedo com a locomotiva, que impossibilita descarregar um torpedo em posição intermediária (T01 na Figura 11) sem descarregar os demais torpedos (T02). Dessa forma, na otimização com associação de requisições e na construção do grafo para solução do VRP, deve-se levar em conta a ordem de entrega dos torpedos.

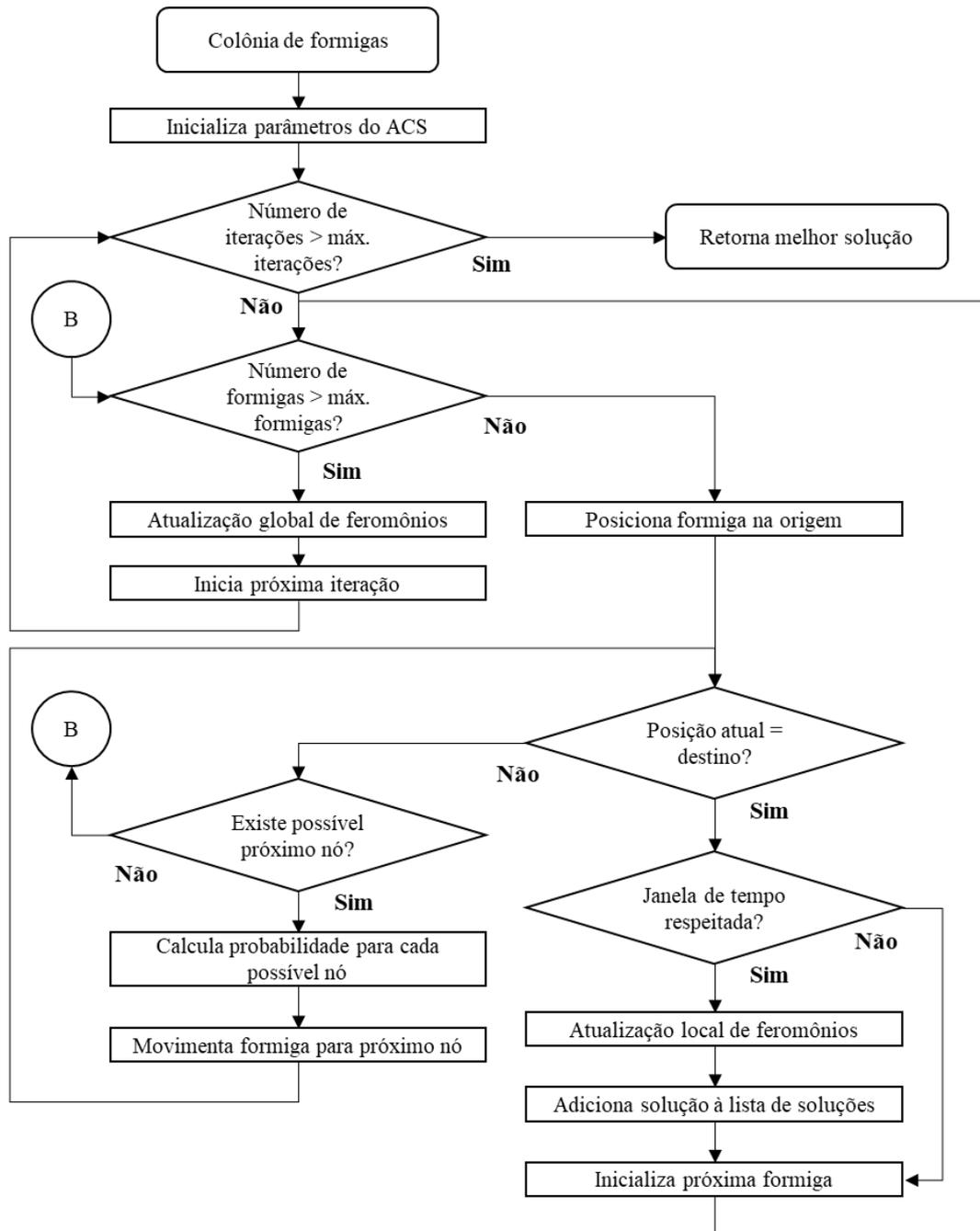
Figura 11 – Disposição de torpedos tracionados por uma locomotiva



Fonte: Autoria própria (2019).

O cálculo do trajeto entre locomotivas e torpedos, apresentado no fluxograma de construção do grafo das soluções (Figura 9), utiliza um algoritmo baseado em colônia de formigas conforme a Figura 12. Baseado na ideia de Stodola e Mazal (2016), uma colônia é criada para cada locomotiva. Em analogia ao problema de roteamento de veículos com múltiplos depósitos, a posição desta se torna um depósito ou garagem virtual, uma vez que no problema proposto a locomotiva não retorna a uma determinada posição após a movimentação. Cada colônia busca a solução do grafo para uma locomotiva, a melhor solução global é selecionada e validada frente às restrições de tempo. Caso alguma restrição seja comprometida, a solução é descartada. No fluxograma da Figura 12 é possível observar as limitações do número de iterações e formigas utilizadas no ACS e também as atualizações locais e globais de feromônios. Caso uma formiga fique sem saída durante a movimentação, ou seja, não há outro nó adjacente disponível que já não tenha sido visitado neste percurso, o algoritmo inicializa a próxima formiga (interconexão “B”).

Figura 12 – Fluxograma do algoritmo de colônia de formigas



Fonte: Autoria própria (2019).

4.2 Modelos de Temperatura

Para a resolução do problema de otimização proposto anteriormente, faz-se necessário conhecer a temperatura interna do carro torpedo e do ferro gusa durante a movimentação deste. Como a temperatura do metal líquido é medida apenas no momento do

carregamento do carro torpedo na planta do alto forno e no início do tratamento do gusa na aciaria (logo após a descarga do carro torpedo), faz-se necessário a estimação desta temperatura durante o transporte e movimentação dos carros torpedo. O modelo de temperatura do torpedo foi dividido em duas fases: ciclo de trabalho a vazio e ciclo de trabalho carregado. O primeiro se dá com o carro torpedo vazio, logo a carga térmica do mesmo concentra-se apenas no corpo refratário do *TLC – Torpedo Ladle Car*, como também é chamado o carro torpedo. Já o segundo caracteriza-se pelo torpedo carregado com uma certa quantidade de ferro gusa, o que aumenta a capacidade térmica do mesmo.

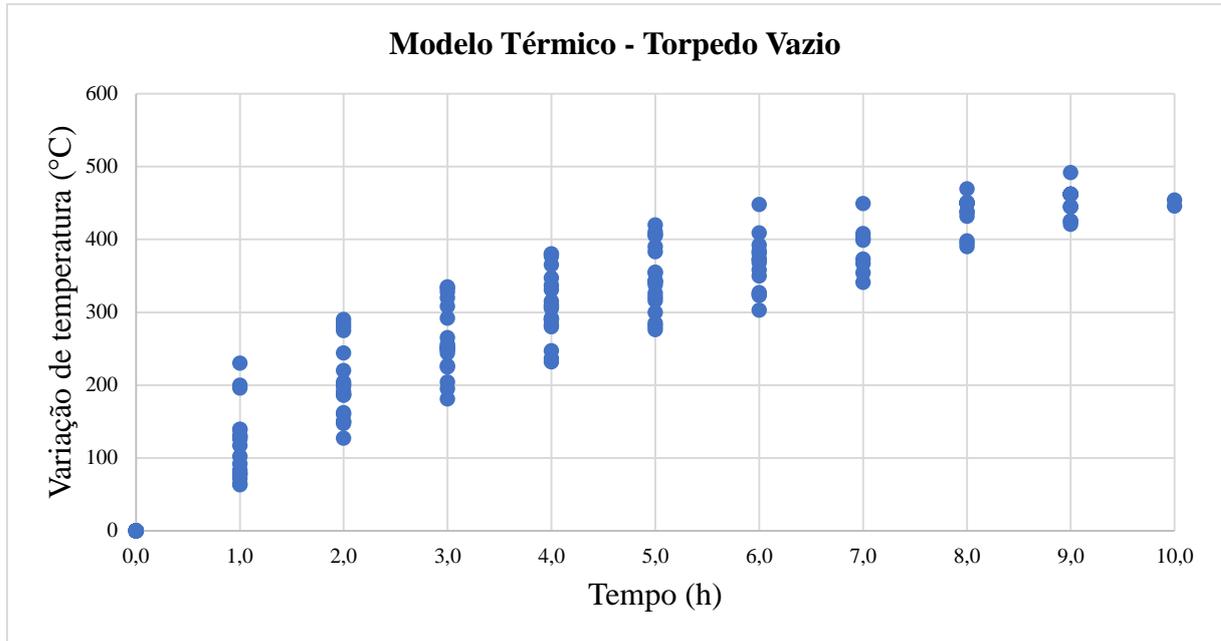
4.2.1 Modelo do Torpedo Vazio

Desconsiderando as condições de desgaste do material refratário do torpedo e assumindo que a pequena variação da temperatura ambiente tem pouca influência no processo de perda de calor, o modelo de temperatura do carro torpedo vazio foi criado baseado: (1) no tempo decorrido, a perda térmica é, conforme os dados levantados, diretamente proporcional ao mesmo, e (2) na temperatura de descarga, que também é diretamente proporcional à queda de temperatura, uma vez que, quanto maior a temperatura inicial, maior o delta de temperatura entre torpedo e ambiente, e conseqüentemente maior velocidade de perda térmica.

Devido à ausência de processo para medição da temperatura do torpedo vazio na chegada da planta do alto forno, o modelo a seguir foi criado com um número reduzido de amostras que foram obtidas através de uma série de medições manuais nos torpedos após a descarga do ferro gusa na aciaria. A primeira medição adotada foi realizada na plataforma da aciaria, logo após a descarga do carro torpedo. Em seguida, o torpedo foi desviado para uma segunda plataforma onde permaneceu sob monitoramento. Outras medições foram realizadas em intervalos regulares de 01 hora para cada torpedo até que sua temperatura atingisse 800°C, temperatura limite de trabalho adotada hoje pela empresa. Ao todo, foram utilizados 19 carros torpedo para a obtenção das amostras de temperaturas.

No Gráfico 1 é possível observar a relação entre a queda de temperatura e o tempo decorrido após a descarga. Destaca-se a queda da inclinação da curva que expressa uma tendência de redução da velocidade de perda de calor, resultante da diminuição da diferença de temperatura do torpedo em relação à temperatura ambiente. Além disso, deve-se observar que a quantidade de amostras diminui ao longo do tempo, uma vez que foram realizadas amostras até atingir 800°C, após este valor os carros eram direcionados novamente ao processo.

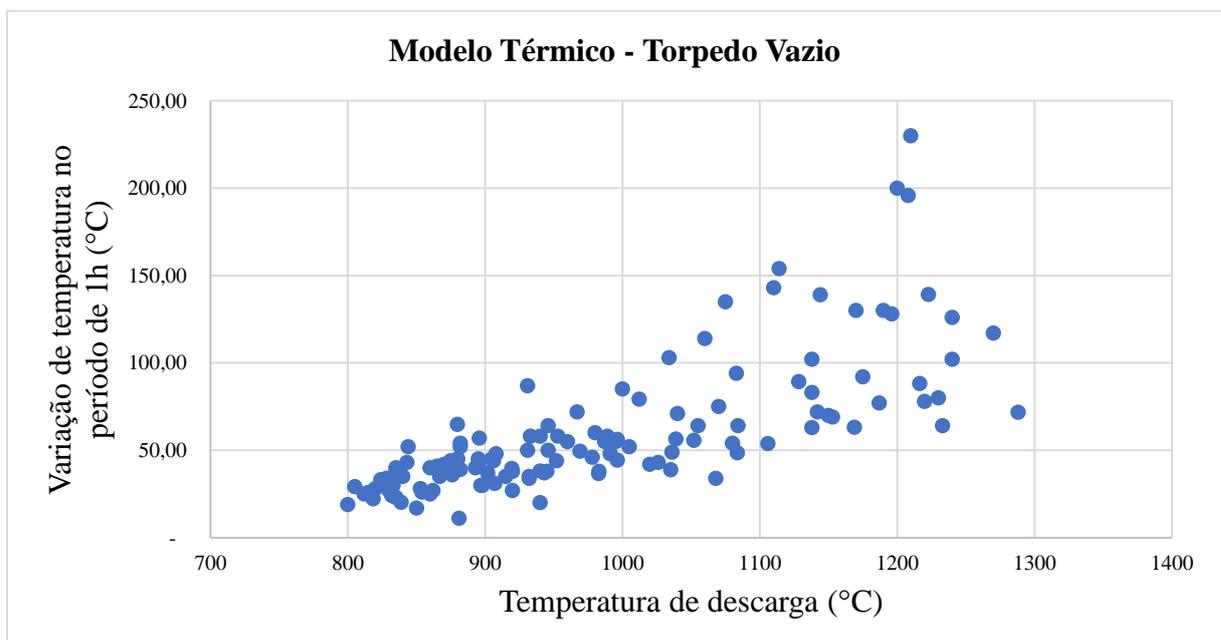
Gráfico 1 – Modelo térmico do TLC vazio: variação de temperatura x tempo



Fonte: Autoria própria (2019).

A relação entre a temperatura de descarga do torpedo e a perda de calor pode ser observada no Gráfico 2. A queda de temperatura representada no Gráfico 2 refere-se ao período de uma hora. Nota-se a relação direta entre as duas variáveis, ou seja, quanto maior a temperatura do torpedo, maior a perda no mesmo período, pois maior é a diferença entre a temperatura do torpedo e a temperatura ambiente.

Gráfico 2 – Modelo térmico do TLC vazio: variação de temp. por hora x temp. de descarga



Fonte: Autoria própria (2019).

O modelo do TLC vazio foi então criado com base na relação entre as variáveis dos Gráficos 1 e 2. Foi escolhido um modelo não linear para a relação do tempo com a queda de temperatura no TLC vazio.

Para estimação dos coeficientes do modelo adotou-se a minimização da raiz do erro médio quadrático (RMSE – *Root Mean Square Error*) com o auxílio da ferramenta Solver do software Microsoft Excel. RMSE é a medida de erro mais comumente utilizada para se medir a qualidade do ajuste de um modelo, e é calculada conforme Equação 4.24, onde N é a quantidade de amostras utilizadas na estimação, y_t é o valor medido e \hat{y}_t é o valor estimado para a amostra t .

$$RMSE = \sqrt{\frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{N}} \quad (4.24)$$

Dessa forma, a aproximação do modelo de temperatura para o carro torpedo vazio é expressa pela Equação 4.25, em que y é a queda de temperatura em graus Celsius, x_1 o tempo decorrido em horas e x_2 é a temperatura inicial do torpedo, também em graus Celsius, ou seja, a temperatura medida do ferro gusa no momento da descarga. Além disso, devido ao baixo número de amostras, o modelo é limitado em relação à temperatura inicial e ao tempo decorrido nos intervalos definidos em 4.26 e 4.27.

$$y = x_1(-4,6075x_1 + 0,0655x_2 + 5) \quad (4.25)$$

Sujeito a:

$$0 \leq x_1 \leq 10 \quad (4.26)$$

$$800 \leq x_2 \leq 1600 \quad (4.27)$$

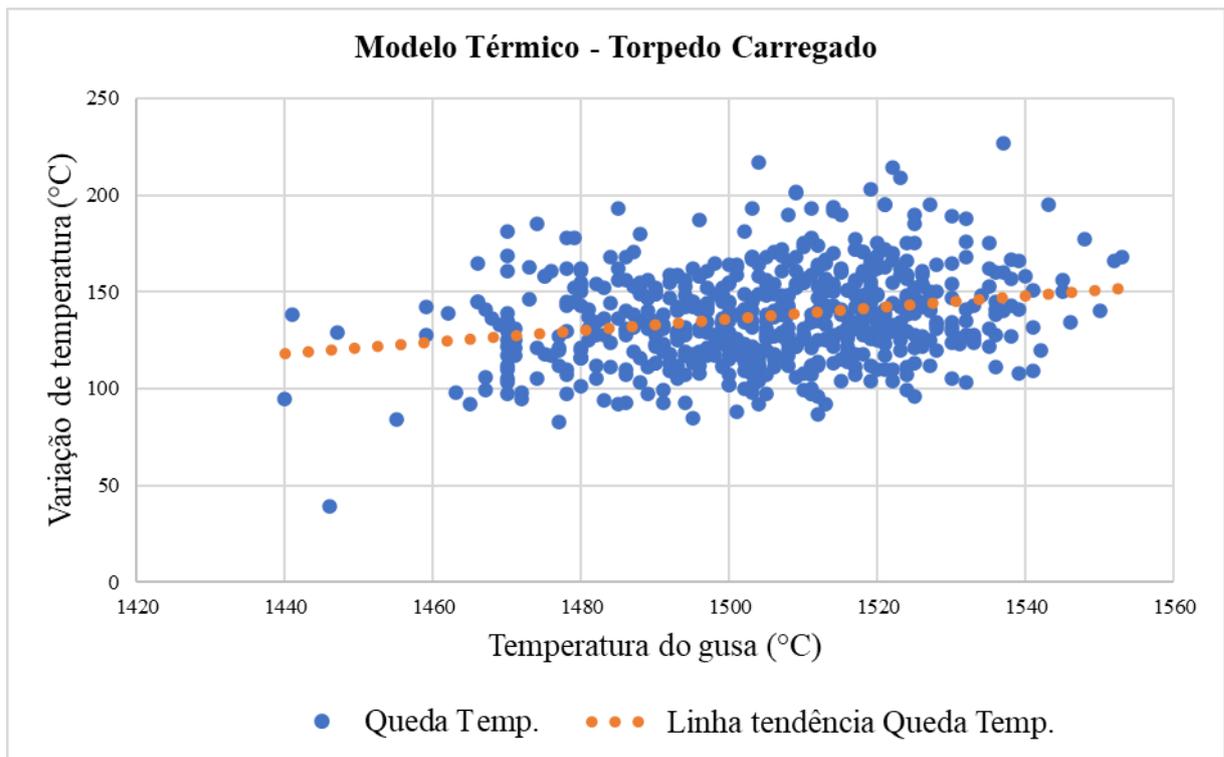
4.2.2 Modelo do Torpedo Carregado

O modelo térmico do carro torpedo carregado de ferro gusa difere do modelo do torpedo vazio devido à mudança da capacidade térmica pela presença do metal líquido no interior do mesmo. O conjunto de dados utilizados para a criação do modelo foi obtido a partir de amostras de 6 meses de operação, que apresentam aproximadamente 70.000 registros. A queda de temperatura foi calculada como a diferença entre a temperatura do gusa carregado (obtida durante o carregamento do torpedo no alto forno) e a temperatura do gusa medida na descarga na aciaria.

Considerando que a quantidade de gusa carregado em um carro torpedo é praticamente a mesma em toda corrida, e que a pequena variação da temperatura ambiente tem pouca influência no processo, pode-se simplificar estas variáveis do modelo e, portanto, tem-se que a queda de temperatura do ferro gusa (e conseqüentemente do torpedo) depende basicamente de três variáveis: (1) da temperatura do gusa carregado no carro torpedo; (2) da temperatura do torpedo vazio no momento do carregamento; e (3) do tempo em que o torpedo permaneceu carregado.

O Gráfico 3 mostra a relação entre a temperatura do gusa carregado e a redução de temperatura. Devido à grande massa de dados, uma linha de tendência foi criada para facilitar a visualização da relação entre as variáveis. Observa-se a relação direta entre as variáveis, ou seja, quanto maior a temperatura do gusa, maior a perda de calor.

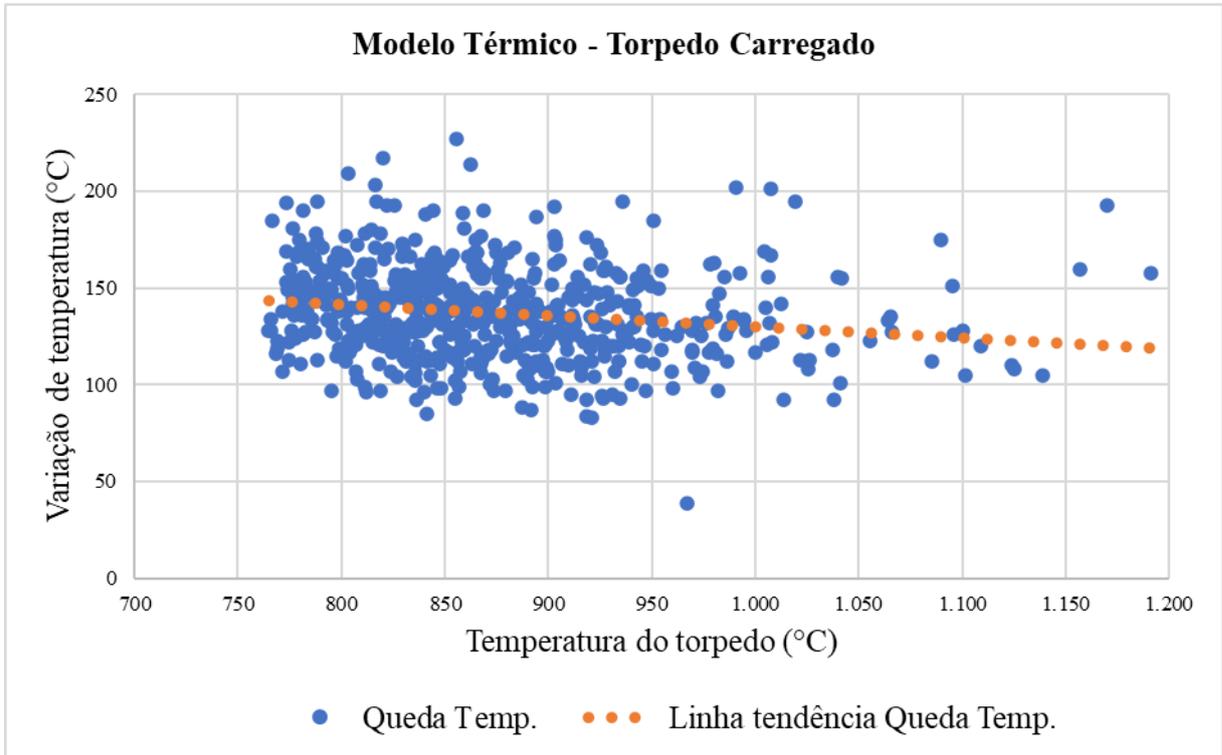
Gráfico 3 – Modelo térmico do TLC cheio: variação de temperatura x temperatura do gusa



Fonte: Autoria própria (2019).

A segunda variável considerada é a temperatura do carro torpedo vazio no momento do enchimento, esta foi estimada a partir do modelo descrito na seção anterior. Esta variável claramente possui relação inversa com a variação de temperatura do torpedo, pois quanto menor a temperatura do torpedo vazio maior a diferença entre a temperatura deste e a temperatura do gusa, e conseqüentemente maior será a perda de calor do gusa para o carro torpedo. Esta relação pode ser vista no Gráfico 4 através de uma linha de tendência para facilitar a visualização.

Gráfico 4 – Modelo térmico do TLC cheio: variação de temperatura x temperatura do TLC



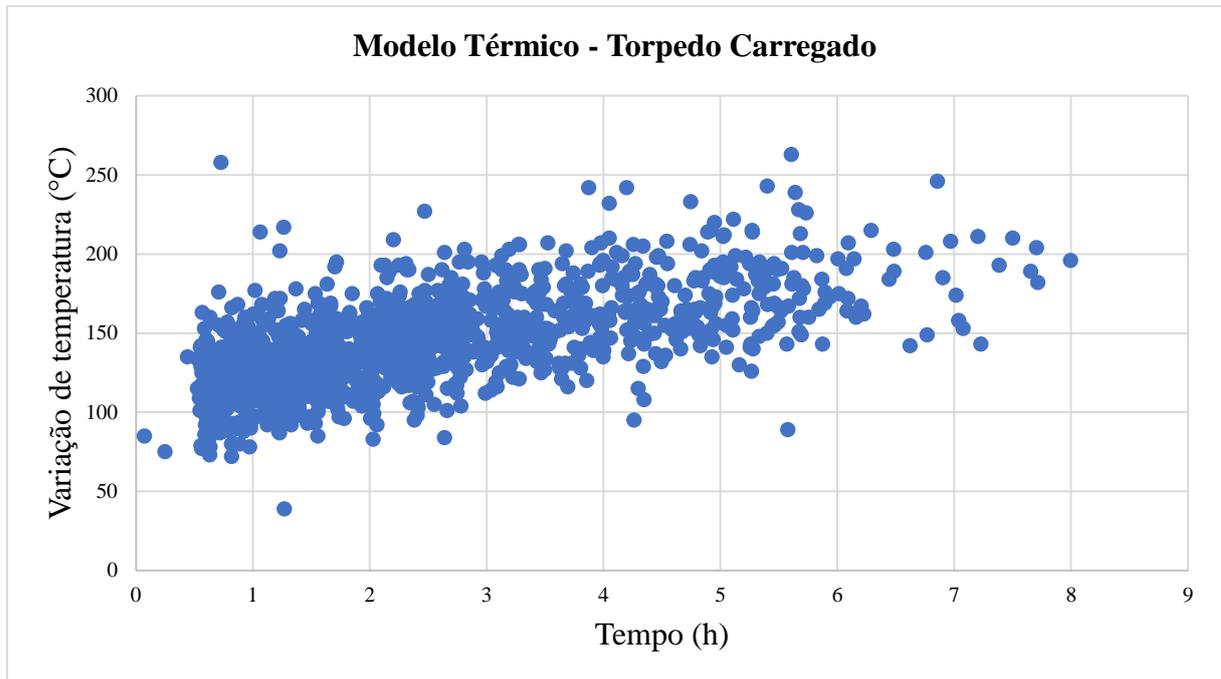
Fonte: Autoria própria (2019).

Por fim, a terceira variável está relacionada ao tempo em que o torpedo permaneceu carregado e, portanto, o tempo em que o sistema torpedo-gusa perde energia térmica para o ambiente.

O Gráfico 5 mostra a relação entre a queda de temperatura e o tempo decorrido. Pela disposição dos dados no gráfico 5, observa-se uma diminuição inicial de temperatura durante o carregamento (com o tempo tendendo a zero) caracterizada pelas perdas de calor do metal líquido: (1) para o canal de corrida; (2) para o ambiente devido à exposição durante o escoamento; e (3) para o próprio carro torpedo, uma vez que o mesmo encontra-se em temperatura mais baixa que o ferro gusa. Além disso, como era esperado, a queda de temperatura é mais acentuada quanto mais tempo o ferro gusa permanece dentro do carro torpedo, neste segundo momento, caracterizada pelas perdas térmicas para o ambiente.

A aproximação do modelo de temperatura para o carro torpedo carregado foi criada com base na relação entre as variáveis dos Gráficos 3, 4 e 5. Foi possível a escolha de um modelo linear pois, diferentemente do modelo de torpedo vazio, existe uma queda de temperatura mesmo para tempo igual a zero. Essa queda de temperatura deve-se à perda de energia térmica do ferro gusa para o canal de corrida durante o carregamento do torpedo. E existe para todo carro torpedo carregado.

Gráfico 5 – Modelo térmico do TLC cheio: variação de temperatura x tempo



Fonte: Autoria própria (2019).

Para estimação dos coeficientes do modelo adotou-se a minimização da raiz do erro médio quadrático (RMSE), também com o auxílio da ferramenta Solver do software Microsoft Excel. Assim, o modelo é expresso pela Equação 4.28, em que y é a queda de temperatura em graus Celsius, x_1 e x_2 representam a temperatura do ferro gusa e a temperatura do carro torpedo vazio, respectivamente, também em graus Celsius, e por fim x_3 é o tempo decorrido após o carregamento em horas. Devido à composição da base de dados e às limitações das variáveis nas amostras da base, estas variáveis também foram limitadas no modelo conforme os intervalos descritos em 4.29 a 4.31. A temperatura do ferro gusa é limitada pelo próprio processo, enquanto a do torpedo vazio é limitada por práticas de preservação da vida útil do material refratário.

$$y = 0,1096x_1 - 0,0575x_2 + 11,7166x_3 - 11,1250 \quad (4.28)$$

Sujeito a:

$$1000 \leq x_1 \leq 1700 \quad (4.29)$$

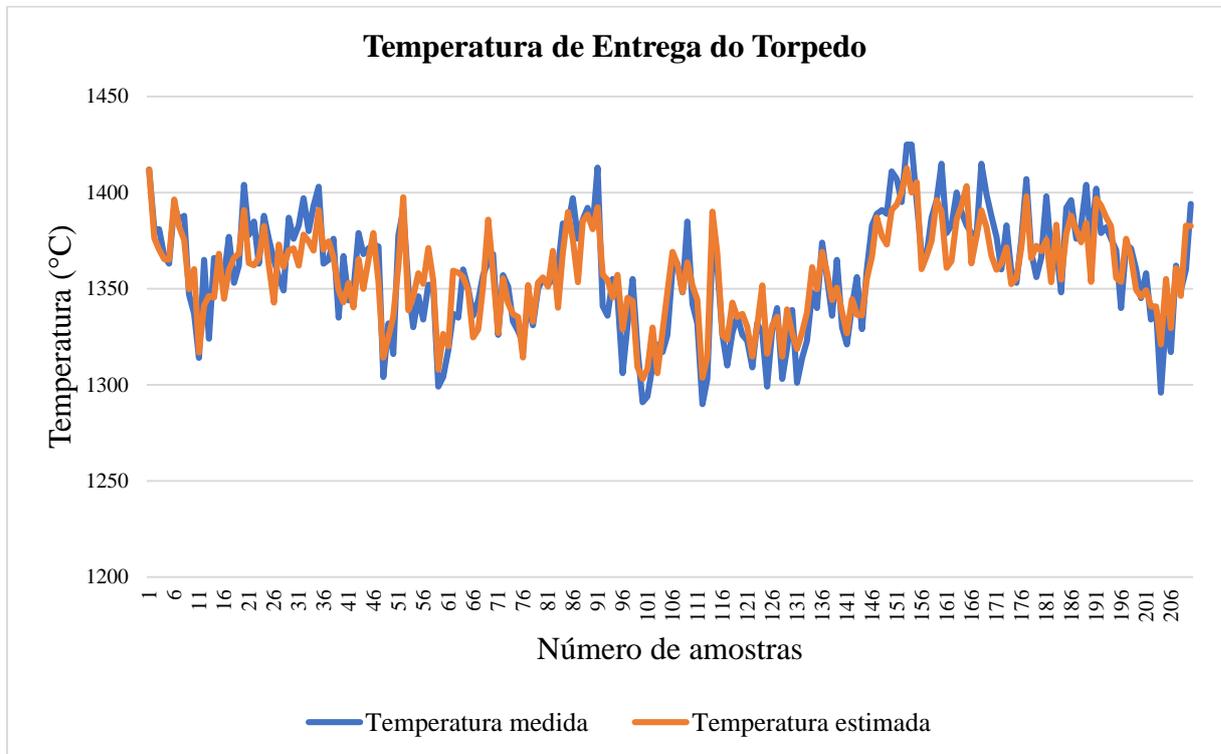
$$800 \leq x_2 \leq 1500 \quad (4.30)$$

$$0 \leq x_3 \leq 12 \quad (4.31)$$

4.2.3 Validação dos Modelos Térmicos

Para a validação do modelo de temperatura do torpedo carregado foi utilizada uma base de dados diferente da utilizada na criação do modelo. Esta base é constituída de um total de aproximadamente 200 registros que representam cerca de 5 dias de operação. O Gráfico 6 mostra a diferença entre a temperatura medida e a temperatura estimada para o número de amostras especificado acima. Foi obtido um erro médio de 8,2% na variação da temperatura estimada, o que representa cerca de 0,8% de erro médio entre os valores absolutos em graus Celsius de temperatura estimada e temperatura medida.

Gráfico 6 – Validação do modelo térmico de torpedo carregado



Fonte: Autoria própria (2019).

Para o modelo térmico do torpedo vazio, optou-se por não realizar a validação devido à falta de amostras para a verificação da aderência. Nas simulações, este modelo foi utilizado principalmente para cálculo da temperatura do carro torpedo vazio no momento do recebimento de gusa na planta do alto forno.

4.3 Validação e Testes do Algoritmo

Devido à especificidade da aplicação e às características do problema, a heurística proposta na Seção 4.1 apresenta a aplicação de um modelo próprio de colônia de formigas voltado para a malha ferroviária com otimização de rotas, escolha de itens para coleta e entrega (carros torpedo), considerando também a variável temperatura do torpedo, através dos modelos apresentados na Seção 4.2.

Isso torna difícil a validação do algoritmo frente a outros problemas genéricos encontrados na literatura, bem como a comparação com os resultados obtidos por outros algoritmos, como os aplicados em otimização do problema de roteamento de veículos, por exemplo. Por esta razão, o método adotado para testes do algoritmo proposto baseou-se na comparação dos resultados alcançados pelo simulador apresentado no Capítulo 5 com um histórico de ações tomadas por operadores experientes, através do sistema de automação existente para controle de tráfego da malha ferroviária da usina em questão.

A comparação foi realizada expondo o algoritmo ao mesmo conjunto de condições para tomada de decisão apresentado aos operadores durante um período contínuo de 24hs de operação. Os resultados e análise destas simulações são apresentados no Capítulo 6, bem como os parâmetros de eficiência e comparação que buscam oferecer indicadores para se comparar a distância percorrida, variação de temperatura e performance do algoritmo.

5 O SIMULADOR DE EVENTOS DA MALHA FERROVIÁRIA DE UMA USINA SIDERÚRGICA

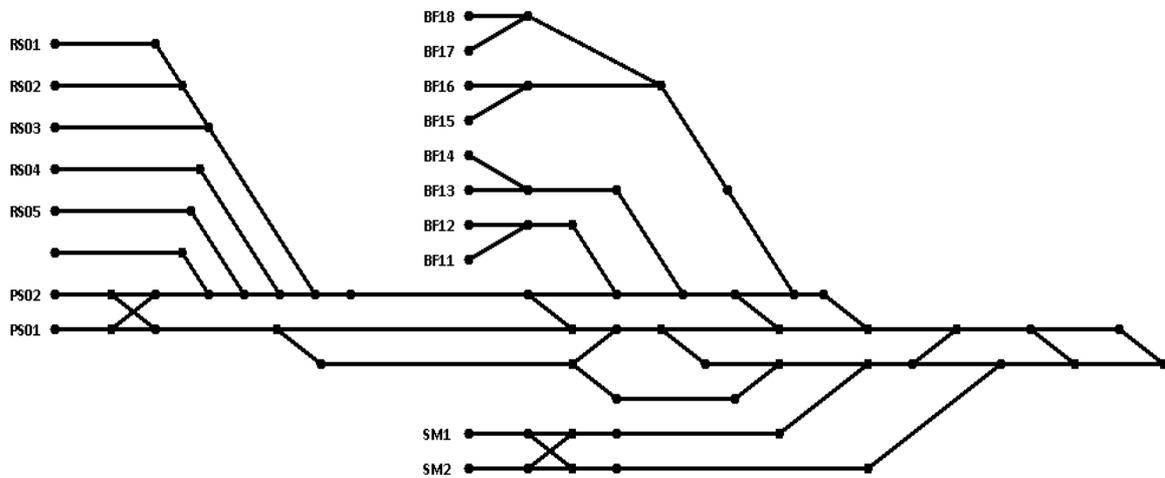
Para realizar os testes necessários para a validação do algoritmo descrito no capítulo anterior, foi utilizado um simulador desenvolvido pelo próprio autor em linguagem de programação Microsoft C#. O simulador tem duas funções básicas: (1) simular o processo físico, gerando dados de corridas, requisições e perdas térmicas com base nos modelos também descritos no Capítulo 4, bem como gerenciar as restrições de tráfego e possibilitar a movimentação dos torpedos e locomotivas; e (2) a segunda função está relacionada à execução da heurística de otimização em si, em que o algoritmo descrito no Capítulo 4 busca a melhor combinação de locomotivas e carros torpedo para o atendimento de cada requisição, considerando a função objetivo descrita pela Equação 3.1.

O modelamento da malha ferroviária utilizada no simulador foi feito através do grafo descrito na Seção 5.1. Na Seção 5.2 são apresentadas as principais características da interface do simulador. A Seção 5.3 descreve as classes e objetos adotados no projeto do simulador, exemplificando seu funcionamento. A Seção 5.4 traz informações importantes sobre como algumas características dos processos de aciaria e alto forno foram incorporadas ao simulador. A Seção 5.5 encerra o capítulo com um resumo sobre as características do simulador desenvolvido, e como estas serão utilizadas no capítulo de resultados para avaliar o algoritmo proposto em comparação com o operador humano.

5.1 Modelamento da Malha Ferroviária

A representação na estrutura de grafo é muito útil na resolução de problemas de roteamento e problemas que envolvam análise combinatória, e por esta razão será utilizada neste trabalho. O grafo foi modelado como ponderado não direcionado, ou seja, suas arestas possuem valores referentes às distâncias entre os pontos interligados e podem ser percorridas em ambos os sentidos. Além disso, ele possui 75 vértices e 85 arestas, sendo baseado na topologia simplificada da malha ferroviária de uma usina siderúrgica integrada. Como trata-se de veículos sobre trilhos, os pontos de origem e destino são trechos da malha ferroviária. Cada trecho foi convertido em um nó, e cada nó foi relacionado aos demais nós através de arestas conforme a conectividade da malha ferroviária, dando origem ao grafo de linha exibido na Figura 13.

Figura 13 – Grafo que representa a malha ferroviária de uma usina siderúrgica



Fonte: Autoria própria (2019).

Os nós indicados pelas *tags* BF11 à BF18 representam os locais de carregamento de carros torpedo do alto forno, e os nós SM1 e SM2, representam os locais de descarga de ferro gusa na aciaria. A quantidade de poços de carregamento e descarga de gusa podem variar conforme o tamanho e a quantidade de plantas de alto fornos e aciarias existentes no complexo. A malha ferroviária acima retrata uma usina com apenas um alto forno com 4 furos de gusa (e consequentemente 8 trechos de recebimento), e uma aciaria com 2 linhas de descarga. Ao lado esquerdo da malha tem-se a oficina de reparo de veículos (RS01 à RS05) e os poços de descarga em emergência PS01 e PS02. Estes poços são utilizados caso o ferro gusa não atenda aos parâmetros de qualidade química da aciaria ou quando há algum problema operacional na aciaria que impossibilite a descarga deste material.

5.2 Interface do Simulador

O modelo contínuo de simulação foi adotado por melhor representar as características do processo. Já a mecânica de avanço de tempo do simulador foi concebida de forma que são realizados passos de tempo fixos de 10 segundos.

Com o objetivo de possibilitar a visualização e o acompanhamento em tempo real da simulação, foi criada uma interface gráfica que pode ser vista na Figura 14. Além do monitoramento, toda configuração e controle do simulador podem ser feitos através desta interface.

Figura 14 – Interface gráfica do Simulador



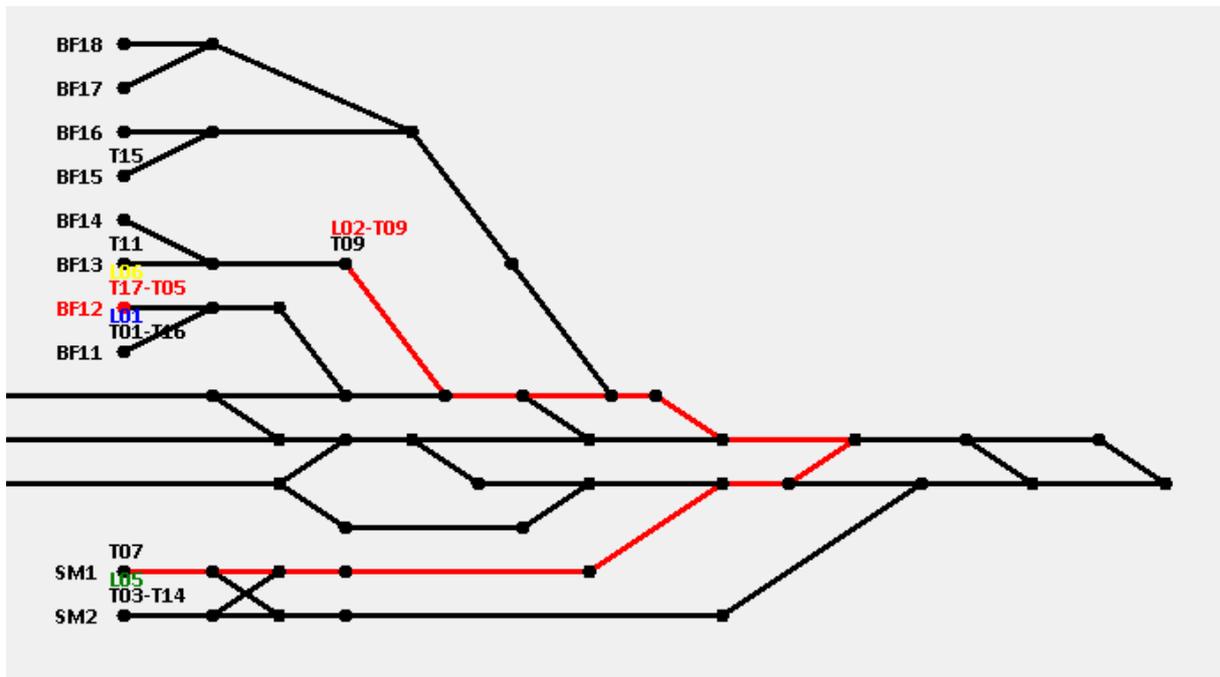
Fonte: Autoria própria (2019).

A barra de estado e controle da simulação é o primeiro quadro marcado em vermelho (1) na Figura 14, e permite visualizar o estado atual da simulação: reiniciar, iniciar, pausar ou parar a simulação. Além disso, na caixa de seleção ao lado direito é possível escolher a velocidade da simulação, de forma que cada minuto em que o simulador permanece em execução corresponde a um valor entre 10 e 10.000 minutos em tempo simulado, ou seja, é possível acelerar a simulação em até 10.000 vezes. O tempo de simulação pode ser visto no segundo quadro (2). Este tempo sempre inicia em zero e é incrementado de acordo com a velocidade de simulação selecionada. O terceiro quadro (3) representa a quantidade total de carros torpedo que estão em reparo, ou seja, não fazem parte do ciclo de simulação. Na usina, base deste estudo, existem 25 carros torpedo e 6 locomotivas, entretanto há necessidade de parada de alguns destes veículos para manutenção. Outro fator a ser levando em consideração é o número de carros torpedo utilizado no ciclo descrito na Subseção 2.3.1. Se este número for muito alto, o percentual de utilização de cada veículo será baixo e, portanto, há maior perda de calor uma vez que o torpedo passará mais tempo vazio. Entretanto, se este número for muito baixo, pode-se comprometer a segurança operacional devido à indisponibilidade de carros torpedo para o carregamento de ferro gusa no alto forno.

No quadro número 4 tem-se uma visão geral da malha ferroviária e de cada um dos veículos, bem como os caminhos percorridos no grafo. Cada locomotiva é desenhada com uma cor distinta, a mesma cor é utilizada para mostrar o caminho planejado a ser percorrido por esta. Na Figura 15, por exemplo, é possível ver a locomotiva L02 (em vermelho) movimentar o carro torpedo T09 do trecho BF13 com destino ao trecho SM1. É possível observar também as locomotivas L01, L05 e L06 paradas nos trechos BF11, SM2 e BF12, respectivamente.

O quadro 5 da Figura 14 mostra o estado das locomotivas e dos torpedos que estão no ciclo. Cada locomotiva pode assumir dois estados: (1) *work* que indica que a mesma está realizando uma movimentação, ou (2) *idle* quando aguarda uma requisição de movimentação. Na lista de torpedos apresenta-se a quantidade de gusa carregada em cada um (em toneladas), o estado e a temperatura estimada no interior do torpedo. O torpedo pode apresentar até três estados: (1) *idle* que representa que o torpedo está parado e disponível para movimentação; (2) *move* que indica que o torpedo está se deslocando; ou (3) *wait* quando o mesmo está posicionado aguardando uma operação de carga ou descarga.

Figura 15 – Interface gráfica do Simulador durante simulação



Fonte: Autoria própria (2019).

O sexto quadro (6) da Figura 14 mostra a relação de corridas do alto forno ou aciaria. Cada corrida é definida pelo horário de início (tempo de simulação), pelo trecho e pelo peso de gusa a ser carregado ou descarregado. O tempo de simulação é expresso no formato “*dd hh:mm:ss*”, em que *dd* representa a quantidade de dias passados a partir do início da simulação,

já os termos *hh*, *mm* e *ss*, representa a quantidade de horas, minutos e segundos, respectivamente, após o início da simulação.

As requisições são as solicitações de carros torpedo cheios ou vazios para atendimento de uma demanda. Elas inicialmente possuem o estado *pend*, que indica que estão pendentes de serem resolvidas. Para cada requisição pendente, o algoritmo de otimização associa uma locomotiva e um (ou dois) torpedos para o atendimento desta. Cada requisição tem seu estado alterado para *work* quando ocorre o início de movimentação da locomotiva e torpedo associados. O terceiro estado que uma requisição pode assumir é o de encerrada (*encer*), e ocorre após a entrega do torpedo no seu ponto de destino. A lista de requisições pode ser vista no quadro 7 da Figura 14.

Por fim, o quadro 8 mostra o histórico de mensagens geradas pelo simulador. As principais mensagens exibidas são de início e término de uma corrida, e o resultado do algoritmo aplicado à uma requisição, indicando inclusive o custo de movimentação calculado.

5.3 Estrutura do Simulador

Para se obter um código mais organizado e estruturado, principalmente devido à complexidade do simulador, optou-se pela utilização de orientação à objetos. Portanto toda simulação é realizada através dos objetos descrito nesta seção. Entretanto para melhor desempenho, o algoritmo em si, apesar de estar contido em uma classe, trabalha com dados de matrizes e vetores, ao invés de lista de objetos, o que representa um ganho considerável de performance. Outra medida adotada para melhorar o desempenho do simulador e do algoritmo foi a opção por processamento paralelo. Aproveitando a tecnologia de processadores com múltiplos núcleos, a separação das rotinas de simulação e de otimização em *threads* distintas, além de torná-las independentes uma das outras, traduz-se em maior performance e melhor aproveitamento dos recursos de hardware disponíveis.

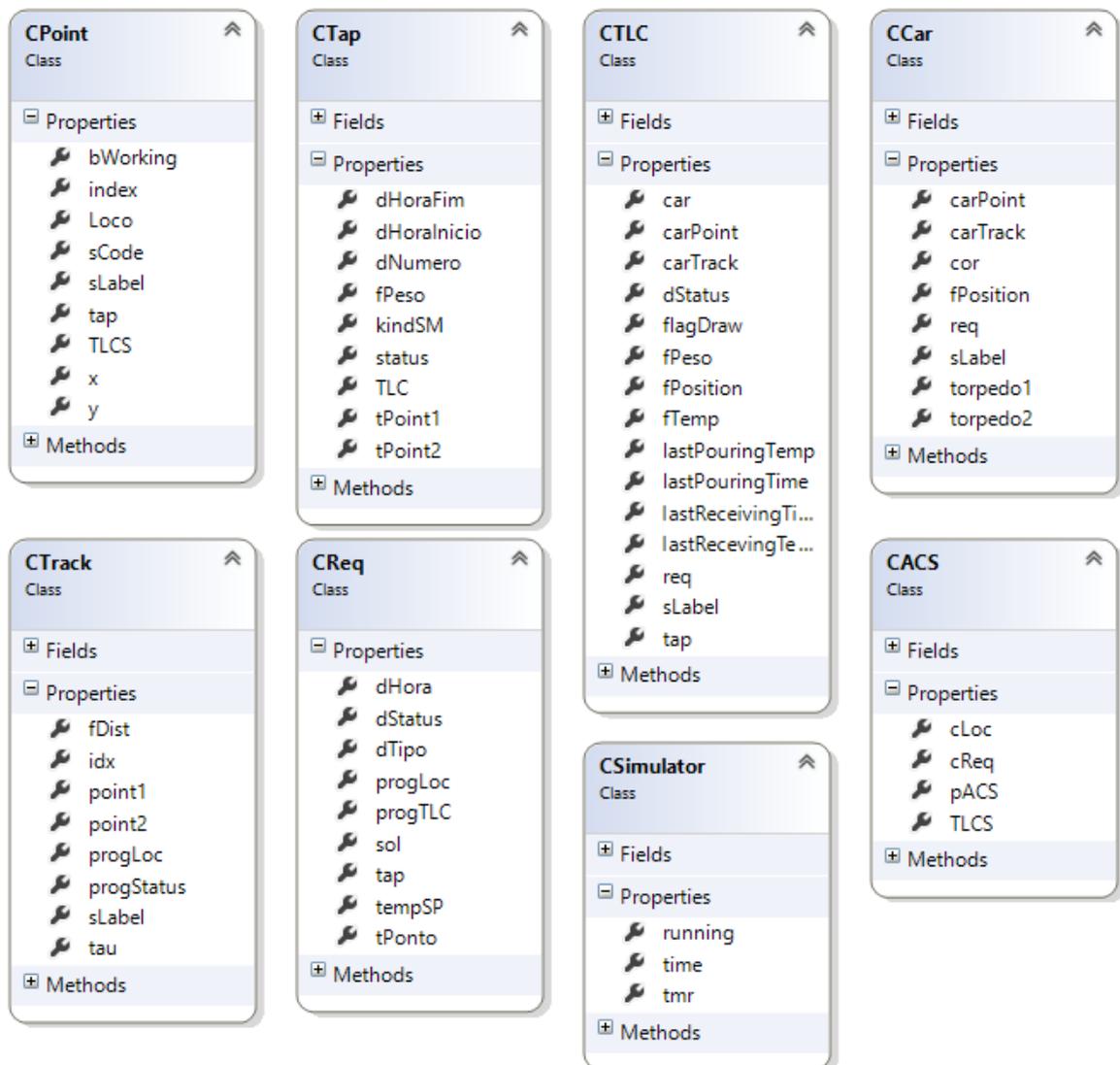
A Figura 16 apresenta as principais classes do simulador. O grafo da malha ferroviária é representado basicamente por duas classes: *CPoint* e *CTrack*. A primeira representa cada ponto do grafo, ou seja, para cada trecho da malha ferroviária há um objeto correspondente nesta classe. Um objeto da classe *CTrack* relaciona dois objetos *CPoint* através dos campos *point1* e *point2*, atribuindo uma distância *fDist* entre eles.

Cada corrida é representada por um objeto da classe *CTap*. Uma corrida da aciaria é diferenciada de uma corrida do alto forno pela propriedade *kindSM*. Cada corrida deve

obrigatoriamente possuir um horário de início, um peso e os dois pontos que representam o furo de gusa do alto forno ou o *pit* da aciaria em que o metal líquido será descarregado. O peso representa a quantidade de gusa a ser carregado para uma corrida do alto forno, ou a quantidade de gusa a ser descarregado em uma corrida da aciaria.

As requisições de movimentação são definidas pela classe *CReq*, e são criadas a partir das corridas existentes. A requisição contém, inicialmente, a hora máxima para que o torpedo esteja disponível no ponto desejado, o trecho *tPonto* para a entrega do torpedo e o tipo *dTipo* de carro solicitado que informa ao algoritmo se é uma requisição de um torpedo vazio para o alto forno, ou de um torpedo carregado para a aciaria. As demais informações são preenchidas pelo próprio algoritmo quando este encontra uma solução factível.

Figura 16 – Classes do simulador



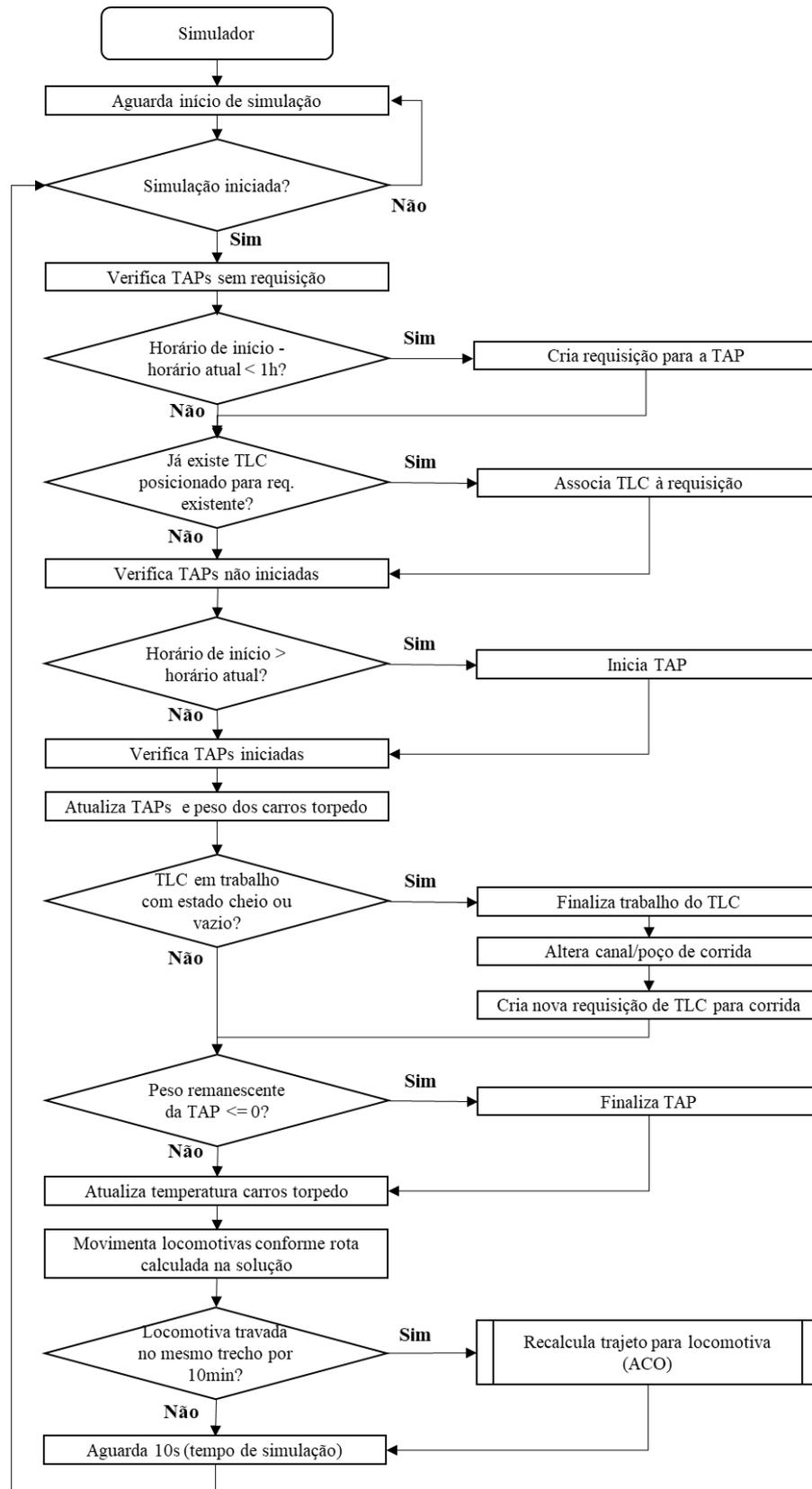
Os objetos das classes *CCar* e *CTLC* representam as locomotivas e carros torpedo, respectivamente. As propriedades *carPoint* e *carTrack* são utilizadas para definição da localização dos veículos, e são modificadas apenas pela classe *CSimulator*. É importante observar que uma locomotiva pode carregar no máximo dois carros torpedo, portanto a propriedade responsável pela associação dos torpedos consiste em apenas *torpedo1* e *torpedo2*. As demais propriedades da classe *CTLC* são responsáveis por armazenar as informações de estado e condição do torpedo, como por exemplo: a temperatura, o peso atual ou o último momento em que foi carregado ou descarregado.

A classe *CACS* é responsável pelo algoritmo de otimização e, a partir de um ponto de destino, calcula a combinação de locomotiva e torpedo que implica em menor custo. Portanto para cada requisição deve-se instanciar um novo objeto desta classe.

Por fim, a classe *CSimulator* tem o papel de gerenciar o simulador. Um objeto desta classe é responsável por toda e qualquer simulação do ambiente. É através dela que uma corrida é iniciada quando o horário de início é alcançado, e é encerrada quando a quantidade de gusa a ser carregada ou descarregada torna-se zero. Ela também é responsável pela criação das requisições que, para simulação das condições reais do processo, só são criadas uma hora antes do início da corrida, tempo estimado em que o operador recebe a solicitação. Dessa forma, o algoritmo só terá conhecimento desta informação após a criação da requisição, assim como o operador no seu dia a dia de trabalho. Além disso, o objeto simulador é quem roda o modelo de temperatura dos carros torpedo e realiza as movimentações das locomotivas conforme solução calculada pelo algoritmo.

O fluxograma que representa o funcionamento do simulador pode ser visto na Figura 17. Após o início da simulação, verifica-se as corridas sem requisição e, caso falte menos de uma hora para o início programado da corrida, as requisições são criadas. Este período de uma hora é aproximadamente o tempo em que o operador tem a confirmação da necessidade por parte das áreas da aciaria ou alto forno. Em seguida, é realizada a verificação de torpedos já alocados no ponto de requisição. Caso haja um torpedo com *status* disponível no ponto de início da corrida (posicionado pela otimização de torpedos, por exemplo) este é associado automaticamente à requisição. Para este caso, não há necessidade de utilização dos algoritmos de otimização e solução de requisição, uma vez que esta requisição já possui torpedo associado. Para as demais requisições, o algoritmo de resolução é responsável por encontrar uma solução factível. Logo em seguida, verifica-se entre as corridas ainda não iniciadas quais atingiram o horário de início, conforme horário programado e tempo decorrido de simulação. As corridas que possuem horário programado maior que horário atual são iniciadas.

Figura 17 – Fluxograma do Simulador



Fonte: Autoria própria (2019).

O próximo passo é a atualização das informações de corrida e peso dos carros torpedo para cada corrida em andamento. Após atualização do peso, verifica-se se o torpedo em trabalho se encontra cheio (em corridas do alto forno) ou vazio (corridas de aciaria). Caso haja algum carro torpedo que tenha terminado a operação de carga ou descarga, o status dele é alterado de “*work*” para “*idle*”. Também é alterado o canal de corrida ou poço de descarga, de forma que, se ainda houver saldo remanescente da corrida (gusa a ser transferido para um torpedo ou faltando gusa para completar uma panela de aciaria), será utilizado o torpedo posicionado no outro canal de corrida ou poço de descarga. Uma nova requisição de torpedo é criada pelo simulador a fim de substituir o torpedo que tenha terminado o trabalho. O peso remanescente da corrida é verificado, e caso seja igual a zero a corrida é finalizada. Em seguida, a temperatura dos carros torpedo é recalculada conforme modelos térmicos de torpedo carregado e vazio apresentados no Capítulo 4.

Por fim, a movimentação destes é feita conforme ordens de movimentação criadas pelo algoritmo de solução. É importante ressaltar que toda movimentação é realizada respeitando as regras de intertravamento e impedimento de uma malha ferroviária real, onde não é possível que dois veículos distintos (não acoplados) ocupem o mesmo trecho simultaneamente para que não haja risco de colisão. Além disso, trechos bloqueados (pela manutenção, por exemplo) também não permitem ocupação. Para isso, uma lógica de anti-colisão foi criada e implementada na rotina de movimentação.

Devido ao dinamismo da malha ferroviária, com as constantes mudanças de disponibilidade e de condições das vias, um algoritmo para recalculer o melhor caminho é necessário durante a movimentação da locomotiva para garantir que a locomotiva não tenha seu percurso bloqueado devido à ocupação de algum trecho por outro veículo ou à interdição de algum trecho para manutenção. Optou-se pela utilização de colônia de formigas para recalculer o trajeto até o destino caso a locomotiva fique estagnada em determinado trecho por um tempo superior a 10 minutos. Os estados que descrevem a malha ferroviária no simulador são atualizados com um passo de 10 segundos.

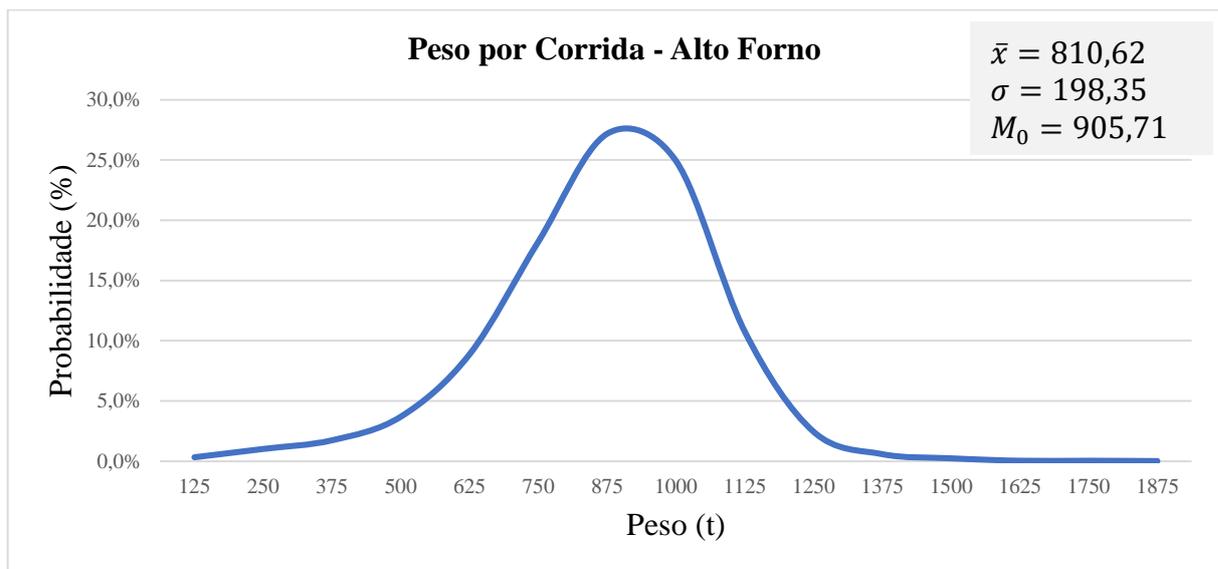
5.4 Simulações do Processo

As características do processo foram incorporadas ao simulador através de distribuições de probabilidade. No total, utilizou-se 3.476 registros de corridas de alto forno e 4.917 registros referentes a corridas de aciaria. A partir deste conjunto de dados históricos que

representam cerca de 6 meses de operação, foi criado um histograma. As classes deste foram definidas conforme conjunto de dados de cada variável. A partir do histograma foram geradas curvas para os eventos de corridas do alto forno e da aciaria. Dessa forma uma corrida de alto forno ou aciaria pode ter seu peso, horário de início e temperatura sorteados aleatoriamente respeitando a distribuição de probabilidade dos gráficos a seguir. Estas características estão representadas nos gráficos 7 a 11.

Estas três características têm influência na dinâmica das corridas do alto forno, entretanto apenas as duas primeiras foram aplicadas à aciaria. A primeira está relacionada ao peso de gusa por corrida. No gráfico 7 observa-se a distribuição do peso total de gusa produzido por corrida do alto forno. Em cada corrida do alto forno produz-se, em geral, entre 625 e 1125 toneladas, que representa o enchimento de 3 a 4 carros torpedo. No canto superior direito de cada gráfico estão representados a média aritmética \bar{x} , o desvio padrão σ e a moda M_0 .

Gráfico 7 – Distribuição do peso de gusa por corrida do alto forno



Fonte: Autoria própria (2019).

Na aciaria, uma corrida representa a quantidade de ferro gusa utilizada em uma panela, que tem aproximadamente a mesma capacidade de um carro torpedo, em torno de 300t. Portanto a variação de peso é bem menor se comparada à uma corrida do alto forno, como pode ser visto no gráfico 8. Através dos gráficos 7 e observa-se que as distribuições de peso do gusa tanto para a aciaria quanto para o alto forno se aproximam muito de uma distribuição normal.

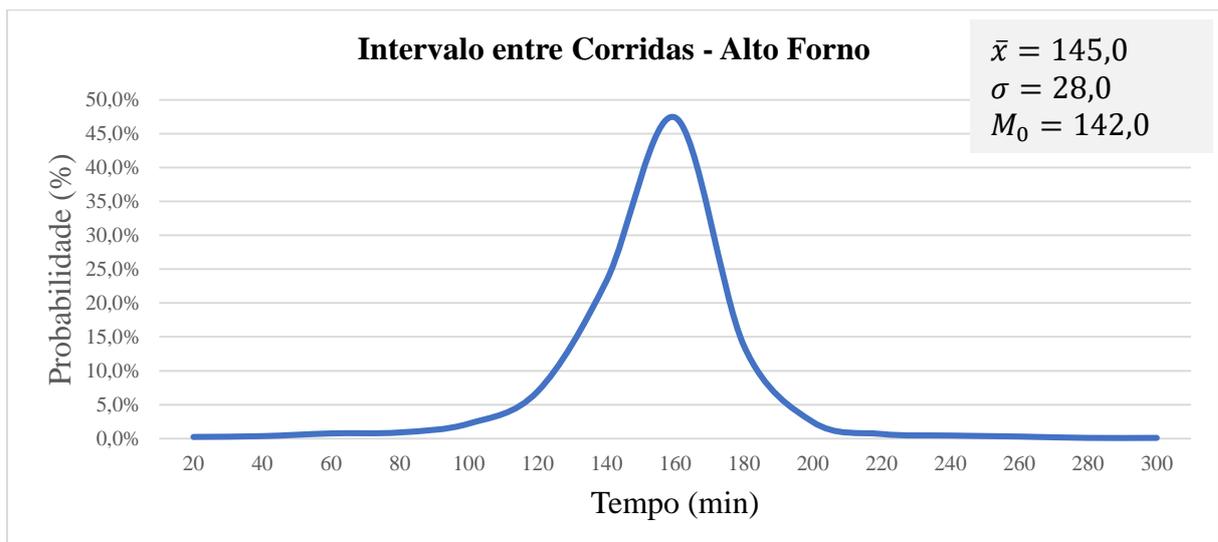
Gráfico 8 – Distribuição do peso de gusa por corrida da aciaria



Fonte: Autoria própria (2019).

A segunda característica comum às duas áreas está relacionada com o intervalo entre o início de corridas, ou seja, indica quanto tempo após o início de uma corrida, a corrida seguinte será iniciada. No gráfico 9 é possível ver esta distribuição para as corridas do alto forno.

Gráfico 9 – Distribuição do intervalo entre corridas do alto forno

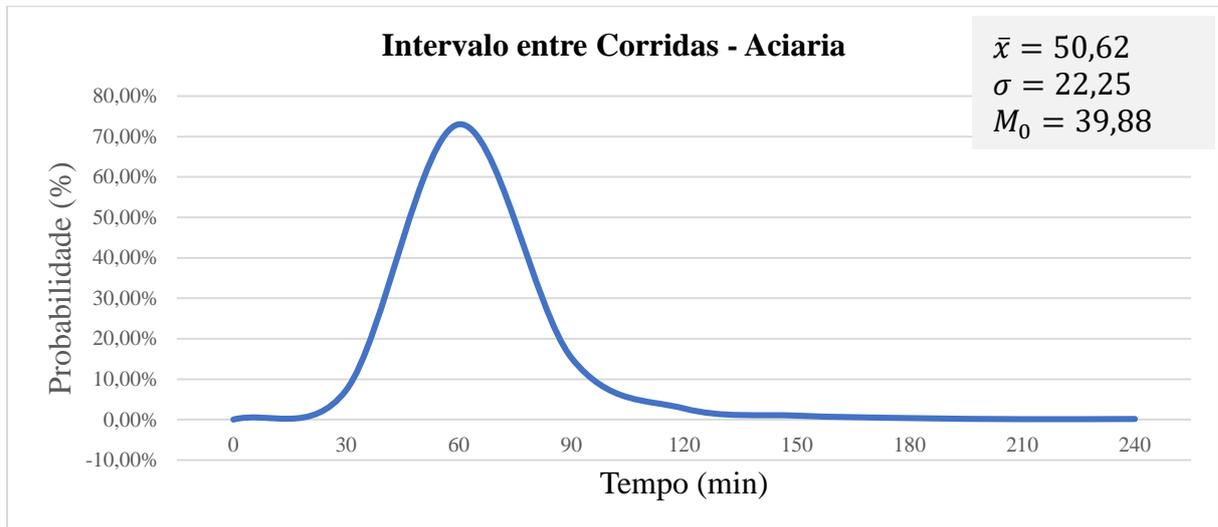


Fonte: Autoria própria (2019).

O gráfico 9 apresenta o intervalo entre o início das corridas da aciaria. Como cada corrida do alto forno manipula, comparativamente, mais gusa que a quantidade utilizada em

uma corrida da aciaria, é esperado que o tempo para manipulação do gusa no primeiro tipo de corrida demande mais tempo e, conseqüentemente, o intervalo entre início de corridas seja maior. Isso fica claro observando os gráficos 9 e 10.

Gráfico 10 – Distribuição do intervalo entre corridas da aciaria

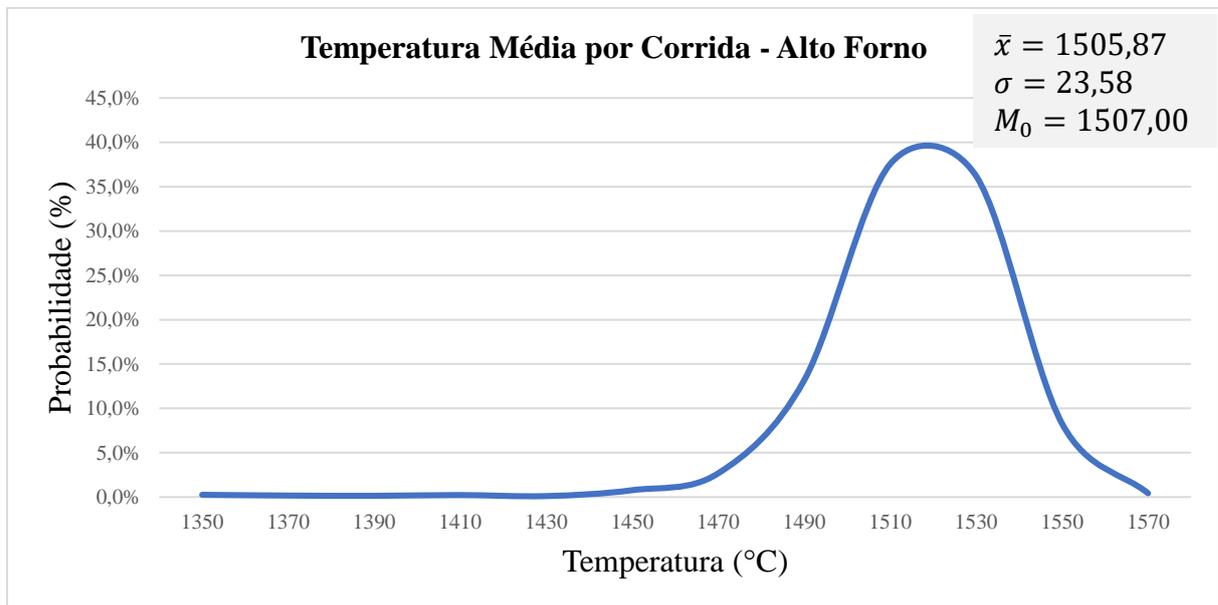


Fonte: Autoria própria (2019).

Por fim, o gráfico 11 apresenta a última variável estimada para as corridas do alto forno: a temperatura média do metal líquido. São realizadas algumas medições de temperatura ao longo da corrida, e a última medição realizada antes do fim de enchimento de um torpedo é adotada como a temperatura do mesmo. Como esta temperatura apresenta uma baixa variabilidade durante o vazamento de uma corrida, optou-se por estimá-la também na forma de distribuição de probabilidade. Esta variável não é estimada para a aciaria, uma vez que a temperatura na aciaria se dá a partir da relação entre a temperatura de saída do alto forno e a perda de calor no transporte do gusa até seu destino.

Devido à pequena variação, outras três características do processo também foram adotadas como constantes e utilizados valores médios. Estes valores podem ser vistos na Tabela 2. A primeira delas é a velocidade de enchimento do torpedo (v_c), que está relacionada com a quantidade de gusa retida no cadinho e a velocidade de escoamento do ferro gusa pelo canal de corrida. Apesar de apresentar uma certa variação durante uma corrida, a adoção desta constante não traz prejuízos à simulação.

Gráfico 11 – Distribuição da temp. média do ferro gusa em uma corrida do alto forno



Fonte: Autoria própria (2019).

A segunda característica tomada como constante é a velocidade de descarga do ferro gusa na aciaria (v_d) e está relacionada ao tempo de transferência do ferro gusa do torpedo para a panela de tratamento. Como esta transferência é realizada através do basculamento do torpedo, ela depende diretamente no ângulo de basculamento utilizado.

Tabela 2 – Constantes adotadas na simulação

Símbolo	Descrição	Valor
v_c	Velocidade média enchimento de torpedo	7,5 ton/min
v_d	Velocidade média de descarga do torpedo	30,0 ton/min
v_t	Velocidade média da locomotiva	40,0 km/h

Fonte: Autoria própria (2019).

Por fim, a terceira variável é a velocidade da locomotiva (v_t). Essa velocidade é praticamente a mesma com a locomotiva tracionando dois carros torpedo ou sem carga, de forma que sua variação se dá apenas nos momentos de partida e parada. Portanto a velocidade dos veículos também foi aproximada por uma constante.

5.5 Informações Adicionais Sobre as Simulações

Tanto a simulação quanto o funcionamento do algoritmo propriamente dito, dependem do grafo apresentado na Seção 5.1. Desta forma, para aplicação do simulador ou do algoritmo proposto para uma malha ferroviária diferente da aqui apresentada, faz-se necessário, primeiro, o modelamento do grafo que represente a configuração da malha em questão.

A Seção 5.2 detalhou a interface do simulador. Esta interface tem como finalidade permitir o acompanhamento da simulação em tempo real, além de permitir configurações adicionais como selecionar o número de carros torpedos no ciclo ou o modo de simulação: com dados provenientes de uma base de dados histórica ou gerados aleatoriamente conforme distribuições de probabilidade apresentadas na Seção 5.4. Além disso, a interface traz informações sobre o histórico de corridas e movimentações dos veículos. Para melhor compreensão do funcionamento do simulador, a estrutura utilizada de classes e objetos foi apresentada na Seção 5.3, bem como o relacionamento entre as classes.

As distribuições de probabilidade apresentadas na Seção 5.4 representam dados reais coletados em campo, e permitem ao usuário realizar simulações que buscam reproduzir situações de uma operação real, e acompanhar o desempenho do algoritmo para diferentes cenários. Além disso, estas distribuições mostram de forma gráfica características de etapas do processo de produção de aço. Como faz-se necessário analisar o desempenho do algoritmo proposto frente ao desempenho de operadores humanos na operação do transporte de ferro gusa, utilizou-se dados do histórico de 24h contínuas de um dia de operação da usina siderúrgica como: horário de início de cada corrida, escolhas de locomotivas e torpedos para atender às solicitações, quantidade de gusa produzida por corrida do alto forno ou consumida em corrida da aciaria. Dessa forma, todas as simulações apresentadas no Capítulo 6 receberam as mesmas requisições de torpedo e poderiam ser comparadas de forma justa. A utilização de dados históricos também foi importante para permitir a comparação da eficiência do algoritmo frente às decisões tomadas por operadores experientes. Neste caso, os resultados dos operadores foram calculados manualmente com base no histórico de movimentações realizadas no dia para atendimento das requisições de carros torpedo.

Os resultados destas simulações e os parâmetros do algoritmo e simulador utilizados para a realização das mesmas, são apresentados no Capítulo 6.

6 RESULTADOS E ANÁLISES

Os resultados apresentados neste trabalho foram obtidos através do cruzamento de dados históricos do gerenciamento da malha ferroviária de uma usina siderúrgica real com os dados obtidos com o simulador descrito no Capítulo 5. Estes resultados foram estruturados de forma a avaliar o atendimento dos objetivos do trabalho: propor um algoritmo para o problema de roteamento de veículos que possa ser aplicado em um sistema ferroviário de alta concentração de manobras, e seja capaz de otimizar uma função objetivo (Subseção 3.1.4) em um curto intervalo de tempo.

As simulações consideram os índices descritos na Seção 6.1 para avaliação, e foram divididas em três etapas. A primeira, apresentada na Seção 6.2, compara resultados obtidos pelo algoritmo com os resultados gerados em campo a partir da ação de operadores frente ao mesmo conjunto de requisições. O intervalo adotado foi de 24 horas de operação contínua, que representa entre 130 e 200 requisições de movimentação. A segunda etapa de simulações, descrita na Seção 6.3, compara o resultado de alterações nos parâmetros de custo da temperatura na função objetivo. E, por fim, a Seção 6.4 mostra a terceira etapa de simulações que analisa a influência de mudanças na quantidade de locomotivas e carros torpedo disponíveis no desempenho da operação do transporte do ferro gusa.

Os parâmetros adotados para o algoritmo de colônia de formigas utilizado nas simulações podem ser vistos na Tabela 3.

Tabela 3 – Parâmetros adotados para o algoritmo

Símbolo	Descrição	Valor
k	Número de formigas	6
n	Número de iterações	80
ρ_1	Taxa de evaporação global de feromônios	0,04
ρ_2	Taxa de evaporação local de feromônios	0,01
α	Parâmetro de influência de feromônio	0,60
β	Parâmetro de influência da qualidade da aresta	1,00
Q	Constante de atualização de feromônio	2.000,00
τ_0	Concentração mínima de feromônio por aresta	0,05
q_0	Parâmetro de escolha entre regra padrão do AS e a regra de maior atratividade (Equação 6.24)	0,90

Fonte: Autoria própria (2019).

Estes parâmetros foram escolhidos com base em Dorigo e Gambardella (1997) e ajustes realizados a partir de simulações realizadas. Eles estão diretamente relacionados às características do problema e do grafo utilizado. Nas simulações realizadas, o número de formigas e alguns parâmetros foram ajustados a fim de obter melhor relação de desempenho e resultados. Por exemplo, foi observado que aumentando o número de formigas não se notava melhora significativa nos resultados. Além disso, uma forma de ajustar os parâmetros do algoritmo é apresentada no artigo do Apêndice A.

Todas as simulações foram realizadas utilizando o mesmo recurso de hardware a fim de se comparar o desempenho em cada etapa de simulação de forma justa. O hardware básico utilizado nas simulações foi um notebook com processador Intel Core i5-3230M de 2,6GHz com 8GB de memória RAM, muito inferior aos servidores utilizados em automação. Para efeitos de comparação, um servidor utilizado no sistema de automação do transporte de gusa da usina siderúrgica em questão possui 4 processadores Intel Xeon E7-4820 de 2,0GHz com 32GB de memória RAM.

6.1 Indicadores de Desempenho

A fim de comparar os resultados apresentados pelo algoritmo e medir o desempenho deste frente aos diferentes cenários de simulação, foram adotados dois tipos de indicadores: de desempenho do processo e de performance do algoritmo.

Os indicadores do primeiro tipo são apresentados na Tabela 4. Tem por finalidade medir variáveis cujo impacto está relacionado ao processo, como distância percorrida e temperatura do ferro líquido. O custo equivalente à distância percorrida d é dado pela soma da distância total, em metros, percorrida por cada locomotiva ao longo da simulação. Através deste é possível medir a eficiência do algoritmo de otimização de carregamento da locomotiva, descrito na Figura 8. Está diretamente ligado à primeira componente da função objetivo dada pela Equação 3.1. Além disso, a quantidade de combustível gasto pelas locomotivas e o desgaste dos equipamentos como locomotiva e trilhos são diretamente proporcionais a este indicador

As temperaturas T_{Fe}^{BF} e T_{Fe}^{SM} representam a temperatura do ferro líquido no momento do enchimento do carro torpedo no alto forno e no momento de descarga na aciaria, respectivamente. Já as temperaturas T_{TLC}^{SM} e T_{TLC}^{BF} , representam a temperatura do carro torpedo no momento da descarga do ferro gusa na aciaria e no momento de chegada do torpedo no alto forno.

Tabela 4 – Indicadores de desempenho de processo

Indicador	Símbolo	Descrição
Distância percorrida	d	Distância total percorrida dada em metros (m)
Temperatura gusa enchimento	T_{Fe}^{BF}	Temperatura do ferro gusa no momento do enchimento do carro torpedo no alto forno (°C)
Temperatura do gusa descarga	T_{Fe}^{SM}	Temperatura do ferro gusa no momento da descarga na aciaria (°C)
Queda de temperatura do gusa	ΔT_{Fe}	Varição de temperatura do ferro gusa durante o transporte (°C)
Temperatura do torpedo descarga	T_{TLC}^{SM}	Temperatura do carro torpedo vazio no momento da descarga de gusa na aciaria (°C)
Temperatura do torpedo no enchimento	T_{TLC}^{BF}	Temperatura do carro torpedo vazio no momento do recebimento de ferro gusa no alto forno (°C)
Queda de temperatura do torpedo	ΔT_{TLC}	Varição de temperatura do carro torpedo vazio
Número de veículos	n_{Loc}	Quantidade de locomotivas disponíveis para realização da movimentação de carros torpedo
Número de torpedos disponíveis	n_{TLC}	Quantidade de carros torpedo disponíveis no ciclo para transporte de ferro gusa
Ocupação da locomotiva	r_{Loc}	Taxa de ocupação da locomotiva (%)

Fonte: Autoria própria (2019).

As variações de temperatura ΔT_{Fe} e ΔT_{TLC} estão relacionadas à terceira e última componente da Equação 3.1, e representam a variação de temperatura do torpedo entre os eventos de carga ou descarga. Este custo indica a variação de temperatura a que foi exposto o refratário do carro torpedo, de forma que a minimização desta variação tem impacto no aumento da vida útil do refratário. Além disso, a diminuição da variação de temperatura do torpedo da saída do Alto Forno até a chegada na Aciaria representa maior temperatura de gusa na Aciaria e, conseqüentemente, menor custo de produção de aço

O número de veículos n_{Loc} representa o custo referente à quantidade de locomotivas utilizadas para movimentação. Está diretamente ligado à segunda componente da função objetivo dada pela Equação 3.1, e relaciona custos ligados à disponibilidade de máquina e mão de obra de maquinista. O número de torpedos disponíveis n_{TLC} representa, como o próprio já diz, a quantidade de carros torpedo disponíveis para o ciclo de trabalho. Quanto menor este número, menor tende ser a variação de temperatura, entretanto, maior tende a ser o atraso das corridas. Por último, para cada locomotiva disponível para a realização de movimentações de torpedo é calculado o tempo que esta passou efetivamente realizando uma movimentação frente ao tempo total de simulação. Este indicador de ocupação da locomotiva

r_{Loc} mostra a ociosidade de cada veículo, auxiliando na tomada de decisão para a diminuição do número de veículos disponibilizados para movimentação.

Os indicadores do segundo tipo tratam do desempenho do algoritmo. Estes indicadores estão descritos na Tabela 5 e avaliam a performance da metaheurística, o esforço gasto para processamento e a aplicabilidade do algoritmo em ambiente industrial.

Tabela 5 – Indicadores de desempenho de algoritmo

Indicador	Símbolo	Descrição
Tempo médio de processamento	\bar{t}_{proc}	Tempo médio gasto para a execução do algoritmo na resolução de cada requisição (s)
Tempo máximo de processamento	t_{proc}^{max}	Tempo máximo gasto para a resolução das requisições (s)
Tempo total de atraso de corridas	t_a	Tempo total de atraso das corridas (min)

Fonte: Autoria própria (2019).

Tempo médio de processamento (\bar{t}_{proc}) representa o tempo médio, em segundos, gasto para a execução do algoritmo e resolução de cada requisição. Está ligado diretamente à eficiência do algoritmo e ao poder de processamento do computador utilizado para as simulações. Visa evidenciar a viabilidade de aplicação deste na otimização de rotas em ambiente de produção. Já o tempo t_{proc}^{max} , mostra o tempo máximo em segundos, gasto pelo algoritmo para a resolução das requisições, evidenciando o tempo máximo necessário para atendimento da requisição mais complexa. É também um indicador importante para se analisar a viabilidade de aplicação deste na otimização de rotas em ambiente de produção.

Por fim, o tempo total de atraso de corridas t_a mostra a soma das diferenças de tempo entre os horários planejados e de início da corrida. De forma a indicar o atraso de cada corrida pela indisponibilidade de carros torpedo. Este valor não representa impacto direto na produção, uma vez que muitas corridas são realizadas em paralelo e estes tempos acabam sendo absorvidos durante o processo de fabricação do aço. Entretanto, a minimização deste atraso é muito importante para o processo, e mostra a eficiência do algoritmo para planejamento e atendimento das demandas de torpedo.

6.2 Simulação 1 – Operador *versus* Algoritmo

Devido à especificidade do problema e à dificuldade de adaptação da função objetivo a outros problemas de otimização, a eficiência do algoritmo foi testada através de uma

comparação do desempenho do algoritmo com o desempenho do operador. Esta comparação permite também a análise dos principais índices e ganhos propostos pelo algoritmo. Para isso, algumas premissas foram adotadas.

Foi selecionado um período de 24 horas de operação, e consideradas apenas movimentações entre alto forno e aciaria. Descartou-se as movimentações para pátio de emergência e oficina de carros torpedo com o intuito de aproximar ao máximo a simulação da operação comum. Assim, estas movimentações foram retiradas também do cálculo de perda média de temperatura dos torpedos cheios e vazios, portanto apenas movimentações diretas entre alto forno e aciaria foram utilizadas para cálculo destes índices.

Apenas o local, horário de corrida e a quantidade de gusa a ser carregado ou descarregado foi passado ao simulador, que ficou responsável por criar as requisições de torpedo. O valor de gusa de cada corrida não é utilizado pelo simulador para calcular a quantidade de torpedos necessários, uma vez que o operador também não possui essa informação durante a operação. As informações sobre local e início de uma corrida são utilizadas pelo simulador para criação das requisições apenas uma hora antes do início da mesma. Esta premissa foi adotada com o intuito de nivelar as condições entre operador e simulador, uma vez que o primeiro só toma conhecimento do correto local de início de uma corrida cerca de uma hora antes do horário de início da mesma.

Por fim, um comparativo de resultados entre os dados históricos operacionais e uma média de 5 simulações realizadas pode ser visto na Tabela 6.

Tabela 6 – Simulação 1: Resultados de desempenho de processo

Tipo	d (m)	T_{Fe}^{BF} (°C)	T_{Fe}^{SM} (°C)	ΔT_{Fe} (°C)	T_{TLC}^{SM} (°C)	T_{TLC}^{BF} (°C)	ΔT_{TLC} (°C)
Operador	448.264,00	1.509,77	1.331,74	178,03	1.330,80	1.031,33	299,47
Algoritmo	371.299,90	1.505,20	1.371,60	133,60	1.345,03	1.109,45	235,58
Variação Alg./Oper.	0,83	1,00	1,03	0,75	1,01	1,08	0,79

Fonte: Autoria própria (2019).

Em ambos os casos foram utilizados 12 torpedos no ciclo e 4 locomotivas em trabalho. A primeira linha apresenta os resultados extraídos de um dia de operação normal com as manobras feitas pelos operadores experientes e treinados no sistema de controle da malha ferroviária. A segunda linha, traz os resultados obtidos através do algoritmo de otimização proposto e a terceira linha, a razão entre os dois resultados. As cinco simulações foram

realizadas para o mesmo conjunto de corridas e requisições solicitadas ao operador. O número de simulações adotadas tem a intenção de diminuir a influência, no resultado, da componente aleatória do algoritmo de otimização por colônia de formigas. O desvio padrão dos resultados das simulações do algoritmo é apresentado na próxima seção, na Tabela 10.

Observa-se uma redução de aproximadamente 17% na distância total percorrida, representando cerca de 76,96 km ao longo de 24hs de operação. A temperatura de enchimento do carro no alto forno é praticamente constante. Existe uma pequena variação desta pois na simulação foram utilizados valores médios por corrida, enquanto que, nas medições reais, cada torpedo possui sua temperatura. Já a quarta coluna, representa a temperatura de descarga na aciaria, ou seja, a temperatura em que o ferro gusa foi entregue na aciaria para tratamento. Esta temperatura teve um aumento de 3,0% em relação aos resultados operacionais, que representa uma redução de 25% da queda de temperatura, ou seja, de 178,03 para 133,60°C. A temperatura média de saída do carro torpedo da aciaria com destino ao alto forno teve um aumento de 15°C, e conseqüentemente a temperatura de chegada no alto forno apresentou um aumento de 8,0%, que representa uma redução da queda de temperatura do carro torpedo vazio de 21,0%, correspondente à 63,89°C.

A Tabela 7 mostra outros indicadores para o algoritmo proposto. Devido à falta de alguns dados e à natureza destes, não é possível calcular estes indicadores com as movimentações comandadas por um operador.

Tabela 7 – Simulação 1: Taxa de utilização da locomotiva e desempenho do algoritmo

Tipo	r_{Loc} (%)				Processamento		t_a (min)
	L_{01}	L_{02}	L_{03}	L_{04}	\bar{t}_{proc} (s)	t_{proc}^{max} (s)	
Operador	-	-	-	-	-	-	-
Algoritmo	0,16	0,20	0,17	0,10	2,08	12,21	16,83

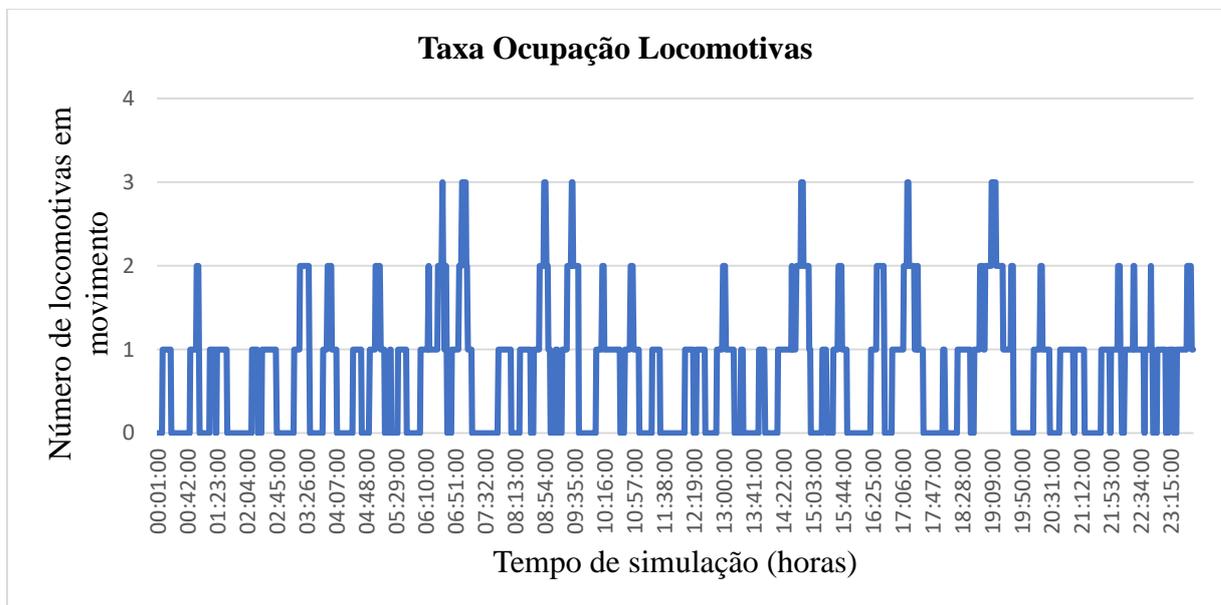
Fonte: Autoria própria (2019).

A taxa de utilização das locomotivas, r_{Loc} (%) indica o percentual do tempo total de 24hs em que a locomotiva permaneceu realizando alguma movimentação. Ou seja, a locomotiva L_{02} , por exemplo, esteve trabalhando na movimentação de torpedos durante 4h e 48m, que equivale a 20% do tempo total de 24h de simulação. O processamento tem por finalidade mostrar o tempo gasto pelo algoritmo para a resolução do problema proposto. A coluna tempo médio mostra que foi necessário em média 2,08s para a resolução de cada requisição de torpedo, e o maior tempo gasto foi de 12,21s, referente à solução mais complexa.

Por fim, a última coluna indica o tempo de atraso das corridas em minutos. Este indicador representa a soma total do tempo de atraso de cada corrida ao iniciar devido à indisponibilidade de carros torpedo. Observa-se que o tempo de 16,83 minutos é relativamente baixo se comparado ao número de 125 requisições geradas nesta simulação. Este indicador não foi calculado para o operador na Tabela 7, pois nos dados reais é disponibilizado apenas o horário efetivo de início das corridas, logo não haveria informações para o cálculo do atraso para os dados reais.

Por fim, o Gráfico 12 mostra a distribuição do uso de locomotivas durante as 24hs de simulação. Através deste pode-se analisar a possibilidade de redução ou a necessidade de aumento da quantidade de locomotivas disponíveis para movimentação.

Gráfico 12 – Simulação 1: Distribuição de utilização das locomotivas ao longo do tempo



Fonte: Autoria própria (2019).

6.3 Simulação 2 – Variação no Custo de Temperatura

Com o intuito de analisar o impacto de mudanças na componente referente ao custo de temperatura no desempenho do algoritmo, algumas simulações foram realizadas para diferentes funções objetivos. As três primeiras variações focam no peso do custo de temperatura frente às demais componentes da função objetivo proposta. As demais têm como base alguns procedimentos operacionais utilizados atualmente para controle do saldo de gusa, em que, por exemplo, é dada preferência a torpedos que estão por mais tempo carregados (para evitar uma queda na temperatura do gusa), como é o caso do modelo de custo número 4.

Como na simulação 1, foram utilizados 12 torpedos no ciclo e 4 locomotivas em trabalho. Um total de seis variações dos custos de temperatura foram criados, como pode ser visto na Tabela 8. O primeiro considera a variação de temperatura conforme a Equação 3.1 ($\gamma = 1$). Nas variações de custo 2 e 3, o fator γ é alterado conforme Tabela 8. Estas alterações no peso γ , tem a intenção de reduzir e aumentar o impacto do custo de temperatura frente aos demais custos da função objetivo. O modelo de custo de temperatura 4, considera a temperatura atual do torpedo como referência de custo de temperatura, com o intuito de favorecer os torpedos que estão por mais tempo parados. Neste caso, quanto menor sua temperatura, menor o custo. A quinta variação do custo de temperatura tem objetivo contrário ao anterior, ou seja, o menor custo relaciona-se diretamente à quanto maior for a temperatura. Por fim, na última variação, o custo de temperatura é considerado nulo para observar o impacto deste.

Tabela 8 – Simulação 2: Custos de temperatura

Custo Temp.	Equação ref. custo de temperatura
Custo 1	$c(T) = T - T_0 $
Custo 2	$c(T) = 0,5 T - T_0 $
Custo 3	$c(T) = 2,5 T - T_0 $
Custo 4	$c(T) = T$
Custo 5	$c(T) = 2000 - T $
Custo 6	$c(T) = 0$

Fonte: Autoria própria (2019).

Os resultados destas simulações são mostrados na Tabela 9. Com base nestes é possível observar que o melhor desempenho frente à distância total percorrida foi referente aos custos 1 e 6, que apresentaram diferença de aproximadamente 5km. Já o pior resultado foi obtido pelo custo 4, cerca de 20km a mais que a média dos demais resultados.

Não é possível estabelecer uma relação direta das variações na função de custo de temperatura com a distância total percorrida por cada variação de custos, uma vez que a distância está mais fortemente ligada à localização geográfica do torpedo e das locomotivas na malha ferroviária. Esta relação pode ser estabelecida apenas na variação 6, onde a função de custo de temperatura é zero, e portanto, não tem influência sobre a função objetivo, dando maior peso à minimização da distância total percorrida.

Tabela 9 – Simulação 2: Resultados

Tipo	d (m)	ΔT_{Fe} (°C)	ΔT_{TLC} (°C)	r_{Loc} (%)	\bar{t}_{proc} (s)	t_{proc}^{max} (s)	t_a (min)
Custo 1	371.299,90	133,60	235,58	0,16	2,08	12,21	16,83
Custo 2	380.056,32	134,06	235,78	0,16	2,51	13,46	18,53
Custo 3	383.917,54	135,79	231,02	0,16	2,17	13,02	6,10
Custo 4	398.403,88	136,47	240,80	0,17	2,13	14,09	6,00
Custo 5	376.353,60	135,34	237,23	0,16	2,18	10,18	11,03
Custo 6	375.538,05	139,71	245,69	0,16	2,11	12,00	13,64

Fonte: Autoria própria (2019).

Como esperado, as perdas térmicas para carro torpedo cheio e vazio foram menores para as variações que possuem o custo de temperatura associado à função objetivo. Os modelos 1 a 3, em que o custo de temperatura visa minimizar a diferença da temperatura (vide Tabela 8), apresentam melhores resultados de perda térmica tanto para o torpedo cheio quanto vazio. O custo 4 prioriza carros com menor temperatura, dessa forma a tendência é de que se observe um aumento da temperatura média dos torpedos disponíveis, uma vez que os de menor temperatura devem ser selecionados mais rapidamente, entretanto a queda de temperatura tende a ser maior. Já o custo 5 prioriza carros com maior temperatura, portanto tende-se a aumentar a temperatura média de entrega dos torpedos, diminuindo a perda de calor, entretanto diminui-se a temperatura média dos torpedos disponíveis nos casos em que a demanda é menor que a produção de ferro gusa. Tem-se esse mesmo efeito para os custos 1 a 3, em que o objetivo é diminuir a queda de temperatura nas movimentações. Este efeito não tem impacto tão significativo uma vez que, em condições normais, o ritmo de consumo de torpedos da aciaria trabalha em sincronismo com o ritmo de produção do alto forno. Por fim, o modelo 6, em que se objetiva minimizar apenas o custo relacionado à distância, apresenta quedas de temperatura mais acentuadas tanto para o ciclo cheio quanto vazio.

A taxa de utilização da locomotiva se mostra baixa para todos os tipos de custo e será melhor discutida na terceira simulação, apresentada na próxima seção. O tempo médio também permanece praticamente constante para cada um dos custos. A variação 4 apresentou o pior desempenho para o tempo máximo de processamento, utilizando 14,09s para a resolução do problema de maior complexidade. Considerado um valor baixo frente ao hardware utilizado na simulação, levando em conta que o processamento do sistema é dividido com simulação e otimização. Observa-se também que este tempo de processamento não contribui diretamente

para o aumento do tempo de atraso de corridas, uma vez que, neste indicador, o custo 4 apresentou o melhor resultado. O atraso máximo totalizado nas simulações foi de aproximadamente 18 minutos, somados ao longo de 125 movimentações, ou seja, uma média de menos de 9 segundos.

Para comparação acerca da variabilidade da média obtida nas cinco simulações realizadas para cada tipo de custo, a Tabela 10 apresenta o desvio padrão para os resultados apresentados na Tabelas 9.

Tabela 10 – Simulação 2: Desvio Padrão para as simulações

Tipo	d (m)	ΔT_{Fe} (°C)	ΔT_{TLC} (°C)	r_{Loc} (%)	\bar{t}_{proc} (s)	t_{proc}^{max} (s)	t_a (min)
Custo 1	1.938,25	2,11	4,74	0,01	0,26	2,25	3,54
Custo 2	2.009,63	2,21	6,85	0,01	0,29	2,04	3,01
Custo 3	1.652,59	3,73	4,87	0,01	0,23	2,38	2,56
Custo 4	1.532,17	3,14	6,01	0,01	0,14	2,82	2,81
Custo 5	1.762,12	1,68	6,11	0,00	0,17	1,31	2,90
Custo 6	1.893,59	2,32	6,52	0,01	0,13	1,49	3,24

Fonte: Autoria própria (2019).

A classificação dos modelos pode ser vista na Tabela 11. Esta foi realizada conforme notas obtidas por cada modelo para os indicadores: (1) distância total d ; (2) queda de temperatura do torpedo cheio ΔT_{Fe} ; e (3) queda de temperatura do torpedo vazio ΔT_{TLC} .

Tabela 11 – Simulação 2: Classificação

Classificação	Tipo	Nota 1 d	Nota 2 ΔT_{Fe}	Nota 3 ΔT_{TLC}	Nota Final (média)
1°	Custo 1	1,000	1,000	0,689	0,896
2°	Custo 2	0,677	0,925	0,676	0,759
3°	Custo 3	0,534	0,642	1,000	0,725
4°	Custo 5	0,814	0,715	0,577	0,702
5°	Custo 4	0,000	0,530	0,333	0,288
6°	Custo 6	0,844	0,000	0,000	0,281

Fonte: Autoria própria (2019).

As notas foram atribuídas de forma que o modelo que apresentou o melhor desempenho recebeu nota 1 no indicador, e o de pior desempenho recebeu nota 0. Os demais receberam notas proporcionais à faixa entre os modelos de pior e melhor desempenho. Para cálculo da nota final, foi utilizada a média das notas de cada indicador. Dessa forma observa-se o melhor desempenho geral para o modelo de custo 1 e o pior desempenho para o modelo 6.

6.4 Simulação 3 – Testando Cenários de Intervenção

A terceira simulação consiste na análise dos parâmetros já apresentados anteriormente frente a mudanças no cenário de simulação. Duas mudanças básicas foram realizadas: (1) variação do número de carros torpedo no ciclo, e (2) variação do número de locomotivas em trabalho. A primeira consiste no aumento e diminuição do número de torpedos disponíveis para recebimento e transporte de ferro gusa. Nas simulações anteriores, foram utilizados 12 carros torpedo. Nesta simulação serão utilizados de 9 a 13 carros a fim de observar os impactos, principalmente relacionados às perdas de calor e atraso. Já a variação do número de locomotivas consiste em realizar simulações utilizando entre 3 e 5 locomotivas disponíveis para manobras, que deverá impactar, principalmente, na taxa de utilização das locomotivas e no atraso das simulações.

As simulações foram realizadas apenas com o modelo 1, que apresentou o melhor desempenho entre as simulações anteriores. O mesmo conjunto de requisições utilizado em simulações anteriores foi usado na simulação. Estes dados representam solicitações de carros torpedo durante um período de 24h de operação. Os resultados mostrados a seguir representam os valores médios de 5 simulações para cada conjunto torpedo/locomotiva, com o intuito de eliminar os impactos de aleatoriedade do algoritmo. A Tabela 12 mostra os resultados obtidos nas simulações para o custo de deslocamento d e taxa de ocupação da locomotiva r_{Loc} . Observa-se que o aumento da quantidade de locomotivas diminui a distância total percorrida, pois diminui o percurso em que a locomotiva trafega sem um torpedo. Outro ponto importante a ser observado é que com o aumento do número de torpedos diminui-se a distância total percorrida pelas locomotivas devido à maior facilidade de otimização de viagens. Ou seja, quanto mais torpedos no ciclo, maior a chance de se encontrar outro torpedo que permita realizar a otimização da viagem. De forma clara, a taxa de utilização das locomotivas segue o mesmo padrão da distância total percorrida, portanto nota-se a maior taxa de utilização para 9 torpedos e 3 locomotivas, e a menor taxa para a simulação com 13 torpedos e 5 locomotivas.

Tabela 12 – Simulação 3: Custo deslocamento e taxa de utilização locomotivas

n_{TLC}	$d(m)$			$r_{Loc}(\%)$		
	$n_{Loc} = 3$	$n_{Loc} = 4$	$n_{Loc} = 5$	$n_{Loc} = 3$	$n_{Loc} = 4$	$n_{Loc} = 5$
9	493.917,89	461.013,97	435.450,40	0,28	0,20	0,15
10	479.990,16	442.756,01	408.942,82	0,27	0,19	0,14
11	439.969,91	395.495,46	376.469,28	0,25	0,17	0,13
12	411.419,89	371.299,90	365.773,02	0,23	0,16	0,13
13	390.400,87	356.819,61	351.030,72	0,22	0,15	0,12

Fonte: Autoria própria (2019).

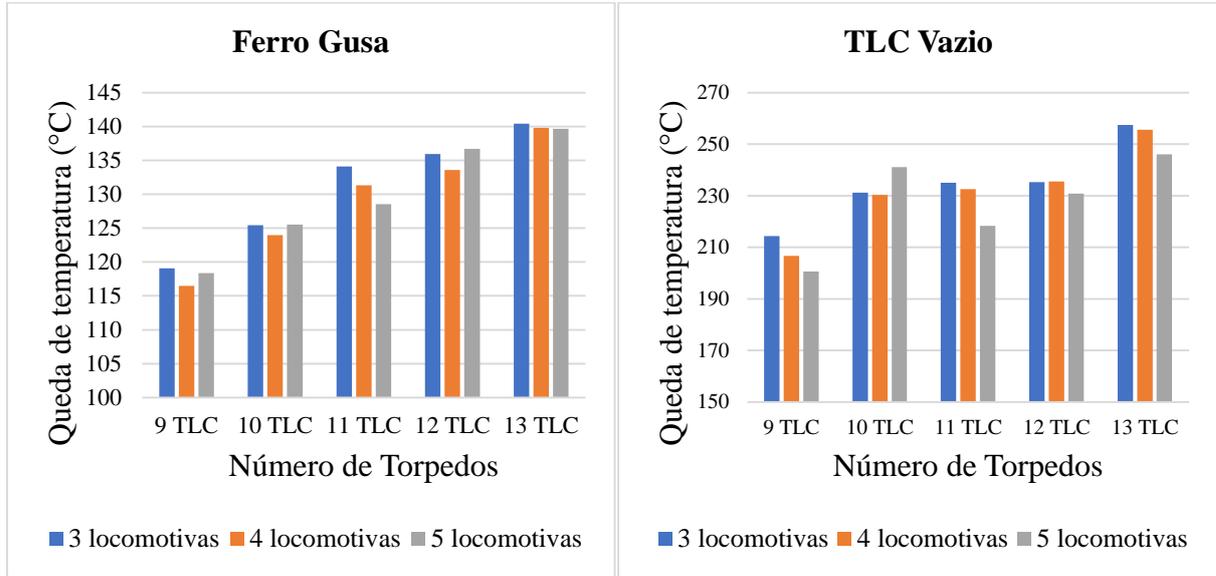
Como pode ser visto na Tabela 13, em alguns casos é possível traçar uma suave tendência de redução das quedas de temperatura com o aumento do número de locomotivas: vide modelos com 9, 11 e 13 torpedos. Isso pode ser justificado pela redução do tempo que se perde aguardando a locomotiva sair da posição de origem e se deslocar até o ponto de coleta do torpedo, após receber uma requisição de movimentação. Uma vez que, quanto maior o número de locomotivas em trabalho, menor a taxa de ocupação das mesmas e maior a probabilidade de diversidade de distribuição geográfica das mesmas pela malha ferroviária.

Tabela 13 – Simulação 3: Queda de temperatura

n_{TLC}	$\Delta T_{Fe} (^\circ C)$			$\Delta T_{TLC} (^\circ C)$		
	$n_{Loc} = 3$	$n_{Loc} = 4$	$n_{Loc} = 5$	$n_{Loc} = 3$	$n_{Loc} = 4$	$n_{Loc} = 5$
9	119,66	117,67	117,09	213,56	205,97	200,86
10	125,40	123,46	127,03	231,85	226,56	238,55
11	134,09	131,34	128,47	234,99	232,53	218,30
12	136,07	133,60	136,38	235,34	235,58	237,18
13	140,43	139,64	139,64	257,46	250,36	246,01

Fonte: Autoria própria (2019).

Entretanto, de uma maneira geral, as quedas de temperatura não sofrem influência significativa pela variação do número de locomotivas. Já com relação ao número de torpedos no ciclo, observa-se uma clara diminuição de perdas com a redução do número de carros torpedo (Tabela 13). Isso deve-se ao fato de que quanto menor a quantidade de carros no ciclo, menor será o tempo de permanência de um torpedo descarregado, e dessa forma menor a perda de calor do torpedo vazio para o ambiente. Os dados da Tabela 13 são exibidos no Gráfico 13 para facilitar a compreensão da influência de cada cenário.

Gráfico 13 – Simulação 3: Queda de temperatura (a) ΔT_{Fe} ; (b) ΔT_{TLC} 

Fonte: Autoria própria (2019).

Através da Tabela 14 é possível observar a performance do algoritmo de acordo com as modificações propostas. Os tempos médio e máximo seguem o mesmo padrão de comportamento. Em geral, estes aumentam conforme o número de locomotivas e torpedos, isso justifica-se pelo número maior de possibilidades a serem analisadas pelo algoritmo. Já o tempo de atraso possui relação inversa ao número carros torpedo, uma vez que quanto menor o número de carros no ciclo, maior tende a ser o atraso.

Tabela 14 – Simulação 3: Indicadores de processamento

n_{TLC}	$\bar{t}_{proc}(s)$			$t_{proc}^{max}(s)$			$t_a(min)$		
	$n_{Loc} = 3$	$n_{Loc} = 4$	$n_{Loc} = 5$	$n_{Loc} = 3$	$n_{Loc} = 4$	$n_{Loc} = 5$	$n_{Loc} = 3$	$n_{Loc} = 4$	$n_{Loc} = 5$
9	1,42	1,68	2,11	7,41	7,43	11,00	31,31	34,73	57,63
10	1,41	2,04	2,14	6,96	8,06	9,33	25,88	29,56	24,75
11	1,45	2,01	2,17	8,20	9,90	10,23	22,75	29,87	33,85
12	1,58	2,08	2,34	9,64	12,21	10,51	14,70	16,83	25,37
13	1,66	1,99	2,63	9,74	13,28	11,66	10,58	13,19	24,36

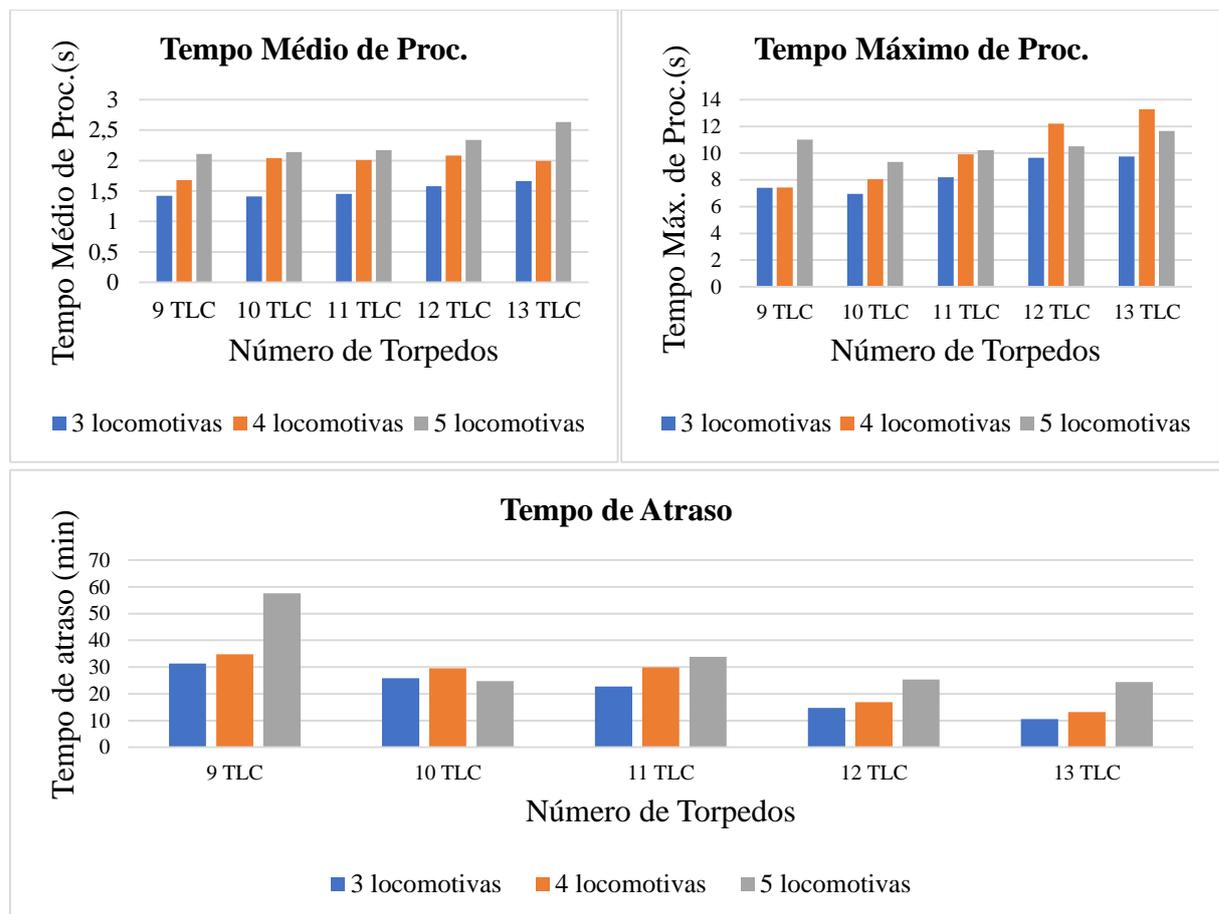
Fonte: Autoria própria (2019).

A relação entre o atraso e o número de locomotivas se dá de forma que quanto maior o número de locomotivas, maior o atraso. Há duas razões para que isto ocorra: a primeira está relacionada ao impacto do aumento dos tempos médio e máximo de processamento. As simulações foram realizadas em velocidade 200:1, ou seja, 1 segundo em tempo real de

processamento representa 200 segundos de simulação. Portanto o tempo máximo de 11,66s de processamento representa cerca de 39min em tempo de simulação, o que poderia gerar um atraso no atendimento da requisição. A segunda razão é que com o aumento da quantidade de locomotivas, há aumento da densidade de tráfego na malha ferroviária e menor número de trechos desocupados, que impacta em atrasos no transporte uma vez que apenas um veículo pode ocupar determinado trecho por vez. Isto deve se agravar caso o número de locomotivas ou torpedos seja muito grande se comparado com a configuração da malha ferroviária. Mesmo que a quantidade de requisições seja mantida, uma maior quantidade de locomotivas possibilita o atendimento de mais requisições simultaneamente.

Os resultados da Tabela 14 são apresentados no Gráfico 14 em forma de barras para facilitar a compreensão dos dados.

Gráfico 14 – Simulação 3: Indicadores de processamento



Fonte: Autoria própria (2019).

Das causas apontadas para o aumento do atraso relacionado ao número de locomotivas, a primeira não ocorreria em ambiente de produção, uma vez que os tempos de processamento médio e máximo são muito pequenos para a obtenção de uma solução. Além

disso, o poder de processamento utilizado é muito limitado se comparado aos sistemas de processamento utilizados na indústria. A segunda causa é inerente ao processo e deverá ocorrer sempre que a demanda por rotas simultâneas aumentar.

7 CONSIDERAÇÕES FINAIS

Neste trabalho, o problema de gerenciamento da malha ferroviária de uma usina siderúrgica foi tratado com: (1) o desenvolvimento de um Simulador de eventos, (2) a modelagem de processos relacionados ao problema, e (3) a apresentação de um algoritmo baseado em Colônia de Formigas para a otimização do gerenciamento de carros torpedo e locomotivas responsáveis pelo transporte de ferro gusa em uma usina siderúrgica integrada. Tanto o controle do tráfego na malha ferroviária quanto a seleção dos torpedos e locomotivas para cada requisição de trabalho são realizados pelo algoritmo.

Uma breve introdução sobre o processo de obtenção do ferro gusa e produção do aço foi apresentada, seguida da dinâmica de funcionamento do transporte de ferro gusa em uma usina siderúrgica, visando dar base ao simulador desenvolvido em linguagem C# (utilizando os conceitos de orientação à objetos), e justificando os processos modelados.

Foi realizado um levantamento do estado da arte da aplicação de otimização por Colônia de Formigas em problemas de roteamento de veículos e apresentados os principais tipos de problemas de roteamento encontrados na literatura. Também foi criado um modelo de roteamento para o problema de transporte de ferro gusa de uma usina siderúrgica. Este modelo trata a otimização da distância percorrida, do número de locomotivas utilizadas no transporte e a redução da variação de temperatura dos carros torpedo.

Foram propostos dois modelos de temperatura para a estimação da temperatura do carro torpedo. O primeiro estima a temperatura do carro torpedo vazio e o segundo contempla a estimação do sistema carro torpedo-ferro gusa para o torpedo carregado de metal líquido.

O algoritmo proposto e os modelos do problema de roteamento de veículos e de estimação de temperatura foram incorporados ao simulador. Onde foi possível realizar diversas simulações a fim de se medir o desempenho e eficiência do algoritmo frente à operação rotineira da equipe responsável pelo gerenciamento do tráfego da malha ferroviária.

Os resultados alcançados pelo algoritmo, comparados às tomadas de decisão de operadores experientes no processo, representaram uma redução de cerca de 77km na distância total percorrida em um período de 24h de operação. Além de uma redução média das quedas de temperatura de 21 e 25% para os torpedos vazios e carregados, respectivamente. Estas reduções contribuem para um aumento de aproximadamente 40°C na temperatura média do ferro gusa entregue na aciaria. A segunda parte das simulações mostrou o impacto de mudanças da influência da temperatura na função objetivo. Nesta foram comparadas diferentes equações e mostrou-se a eficiência da otimização de redução da variação de temperatura frente a simulação

em que o custo de temperatura foi considerado nulo. Por fim, a terceira etapa de simulações abordou os impactos da variação no número de carros torpedo e locomotivas nos resultados de queda de temperatura e distância percorrida. O que permitiu melhor compreensão da dinâmica do processo de transporte de gusa e poderá ser utilizado para simulações de condições operacionais nos mais diferentes cenários.

7.1 Trabalhos futuros

Algumas oportunidades de trabalhos futuros foram observadas ao longo do desenvolvimento desta pesquisa:

1. A primeira, tem por base algumas características do processo que foram incorporadas ao Simulador: informações como o intervalo entre corridas e a quantidade estimada de gusa produzido por corrida podem ser utilizadas para prever o momento e a quantidade de carros torpedo que serão solicitados pelo alto forno, ou que estarão disponíveis para a aciaria. Com esta previsão será possível executar o planejamento de rotas antecipadamente.
2. Um segundo trabalho a ser desenvolvido relaciona-se ao problema dinâmico de roteamento de veículos e consiste em incluir o replanejamento de rotas ao algoritmo atual para rotas já iniciadas. Dessa forma, caso seja verificado a oportunidade de otimização de uma rota já iniciada, uma possibilidade é que esta rota seja replanejada, desde que a nova rota continue atendendo aos requisitos de janela de tempo, não gerando atraso nas corridas.
3. Uma terceira opção para desenvolvimento futuro é o teste do algoritmo e análise do desempenho deste para diferentes topologias de malhas ferroviárias. Poderão ser utilizadas malhas ferroviárias de usinas siderúrgicas espalhadas pelo mundo em que o número de altos fornos e aciarias sofrem alterações. A adaptação do problema a outros modelos conhecidos também deve ser considerada. Uma ideia seria transformar o problema de roteamento de veículos em um problema de alocação dinâmica de veículos, com foco no planejamento e associação de diferentes requisições de transporte.
4. Outra opção é a utilização de outros algoritmos, como o algoritmo de Dijkstra, para calcular a distância entre dois pontos da malha.
5. Para finalizar, apesar da especificidade do problema tratado neste trabalho, a adaptação do algoritmo para outros tipos de problemas de planejamento também pode ser considerada.

REFERÊNCIAS

- ABDALLAH, A. M. F. M.; ESSAM, D. L.; SARKER, R.A. On solving periodic re-optimization dynamic vehicle routing problems. **Applied Soft Computing**, v. 55, p. 1–12, 2017.
- ADEWUMI, A. O.; ADELEKE, O. J. A survey of recent advances in vehicle routing problems. **International Journal of System Assurance Engineering and Management**, v. 9, n. 1, p. 155–172, 2018.
- ASMAR, D.; ELSHAMLI, A.; AREIBI, S. **A Comparative Assessment of ACO Algorithms Within a TSP Environment**. Conference Paper in Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms, 2005.
- BRAEKERS, K.; RAMAEKERS, K.; NIEUWENHUYSE, I. V. The vehicle routing problem: State of the art classification and review. **Computers & Industrial Engineering**, v. 99, p. 300–313, 2016.
- BERBEGLIA, G., CORDEAU, J.-F., GRIBKOVSKAIA, I., LAPORTE, G. Static pickup and delivery problems: A classification scheme and survey. **TOP**, v. 15, n. 1, p. 1–31, 2007.
- BRÄYSY, O.; GENDREAU, M. Vehicle Routing Problem With Time Windows, Part II: Metaheuristics. **Transportation Science**, vol. 39, n. 1, p. 119–139, 2005.
- BRITO, J.; MARTÍNEZ, F.J.; MORENO, J.A.; VERDEGAY, J.L. An ACO hybrid metaheuristic for close–open vehicle routing problems with time windows and fuzzy constraints. **Applied Soft Computing**, v. 32, p. 154–163, 2015.
- BULLNHEIMER, B.; HARTL R.F.; STRAUSS, C. A new rank based version of the ant system: a computational study. **Central European Journal Operations Research and Economics**, 1997.
- BULLNHEIMER, B.; HARTL R.F.; STRAUSS, C. **Applying the ant system to the vehicle routing problem**. In: 2nd International Conference on Metaheuristics, 1997.
- BULLNHEIMER, B.; HARTL R.F.; STRAUSS, C. An improved ant system algorithm for the vehicle routing problem. **Operations Research**, v. 89, p. 319–328, 1999.
- CHENG, A.; YU, D. **Genetic algorithm for vehicle routing problem**. In: Proceedings of the 4th International Conference on Transportation Engineering, p. 2876–2881. Chengdu, China: ICTE, 2013.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. **Operations Research**, v. 12, n. 4, p. 568–582, 1964.
- DANTZIG, G; RAMSER, J. The Truck Dispatching Problem. **Management Science**, v. 6, p. 80-91, 1959.

DAHL, O. J.; MYHRHAUG, B.; NYGAARD, K. **Some features of the SIMULA 67 language.** *In: Proceedings of the second conference on Applications of simulations, Winter Simulation Conference*, p. 29–31, 1968.

DEUTSCH, L. P.; GOLDBERG, A. **Smalltalk yesterday, today and tomorrow.** *BYTE* 16, p.108, 1991.

DHAWAN, C.; NASSA, V.K. Review on Vehicle Routing Problem using Ant Colony Optimization. **International Journal of Advanced Research in Computer Science**, v. 5, n. 6, 2014.

DORIGO, M., **Optimization, Learning and Natural Algorithms**, Tese de Doutorado – Politecnico di Milano, Itália, 1992.

DORIGO, M.; MANIEZZO, V.; COLONI, A. The ant system: optimization by a colony of cooperating ants. **IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)**, v. 26, p. 29-41 (1996).

DORIGO, M.; GAMBARDILLA, L. M. Ant colony system: a cooperative learning approach to the traveling salesman problem. **IEEE Transactions on Evolutionary Computation**, v. 1, p. 53-66 (1997).

DORIGO, M.; DI CARO, G. **Ant colony optimization: a new metaheuristic.** *In: Proceedings of the 1999 Congress on Evolutionary Computation*, v. 2, p. 1470-1477 (1999).

EUCHI, J.; YASSINE, A.; CHABCHOUB, H. The dynamic vehicle routing problem: Solution with hybrid metaheuristic approach. **Swarm and Evolutionary Computation** v. 21, p. 41–53, 2015.

FERREIRA, V. R. A.; BRAGA, A. P. S. **Um Sistema Fuzzy para Ajuste Dinâmico dos Parâmetros de um Sistema de Colônia de Formigas Aplicado ao Controle Ferroviário de uma Usina Siderúrgica.** *In: Proceedings of V Congresso Brasileiro de Sistemas Fuzzy*, p. 397–408, 2018.

GEERDES M., VLIET C.V.D., TOXOPEUS H., MEDEIROS F.T.P. **Práticas Modernas para Operação de Altos-fornos.** São Paulo: Editora Universidade Delft, 2007.

GEERDES M., CHAIGNEAU, R.; KURUNOV, I.; LINGIARDI, O.; RICKETTS J. **Modern Blast Furnace Ironmaking: an introduction.** 3 ed. Netherlands: IOS Press, 2015.

GENDREAU, M.; POTVIN, J.-Y.; BRÄYSY, O.; HASLE, G.; LOKKETANGEN, A. Metaheuristics for the vehicle routing problem and its extensions: A categorized bibliography. The vehicle routing problem: Latest advances and new challenges. **Operations research/computer science interfaces series**, v. 43, p. 143–169, 2008.

GENDREAU, M., TARANTILIS, C. D. **Solving large-scale vehicle routing problems with time windows: The state-of-the-art.** Montreal, 2010.

GHOSH, A., CHATTERJEE, A. **Ironmaking and Steelmaking: Theory and Practice.** 1 ed. Nova Delhi: Rajkamal Electric Press, 2008.

GOLDBARG, M. C.; LUNA, H. P. **Otimização combinatória e programação linear**. 2 ed. Rio de Janeiro: Elsevier, 2005.

HARARY, F. **Graph Theory**. Addison–Wesley, Reading, MA, 1969.

HUANG, R. H; YU, T. H. An effective ant colony optimization algorithm for multi-objective job-shop scheduling with equal-size lot-splitting. **Applied Soft Computing**, v. 57, p. 642–656, 2017.

IRNICH, S. Multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles. **European Journal of Operational Research**, v. 122, n. 2, p. 310-328, 2000.

KACHITVICHYANUKUL, V.; SOMBUNTHAM, P.; KUNNAPAPDEELERT S. Two solution representations for solving multi-depot vehicle routing problem with multiple pickup and delivery requests via PSO. **Computers & Industrial Engineering**, v. 89, p. 125–136, 2015.

KILBY, P.; PROSSER, P.; SHAW, P. **Dynamic VRPs: a study of scenarios**. Technical Report APES-0-1998, University of Strathclyde, 1998.

MAGNANI, R. et. al. **Redução da Perda Térmica em Carro Torpedo**. In: Proceedings of 2002 Congresso Brasileiro de Engenharia e Ciências dos Materiais, p. 511-518.

MAZUNDAR, D., EVANS, J. W. **Modeling of Steelmaking Processes**. 1 ed. Nova Iorque: CRC Press, 2011.

MOLENBRUCH, Y.; BRAEKERS, K.; CARIS, A. Typology and literature review for dial-a-ride problems. **Anais Operations Research** v. 259, p. 295–325, 2017.

NIIR, B. C. E. **The Complete Technology Book on Hot Rolling of Steel**. 2 ed. Delhi: Niir Project Consultancy Services, 2010.

OKULEWICZ, M.; MANDZIUK, J. **Application of Particle Swarm Optimization Algorithm to Dynamic Vehicle Routing Problem**. In: Proceedings of International Conference on Artificial Intelligence and Soft Computing, p. 547–558, 2013.

PINHO, E. G. **Uma Linguagem de Programação Paralela Orientada a Objetos para Arquiteturas de Memória Distribuída**. 2012, 78 f. Dissertação (Mestrado em Ciência da Computação) – Pró-Reitoria de Pesquisa e Pós-Graduação, Universidade Federal do Ceará, Fortaleza, 2012.

SCHRAMA F.; BERG B. Comparison of Kanbara reactor, magnesium mono-injection and lime: magnesium co-injection for hot metal desulfurization. **Millennium Steel India**, v. 4, n. 1, p. 26–31, 2014.

SOMBUNTHAM, P.; KACHITVICHYANUKUL, V. **Multi-depot Vehicle Routing Problem with Pickup and Delivery Requests**. In: Proceedings of AIP Conference 1285 71, 2010.

STODOLA, P; MAZAL, J. Applying the ant colony optimisation algorithm to the capacitated multi-depot vehicle routing problem. **International Journal of Bio-Inspired Computation**, v. 8, n. 4, p. 228–233, 2016.

STODOLA, P. Using Metaheuristics on the Multi-Depot Vehicle Routing Problem with Modified Optimization Criterion. **Algorithms**, v. 11, n. 74, 2018.

STUTZLE, T.; HOOS, H. H. Max-Min ant system. **Future Generation Computer Systems**, v. 16, p. 889-914, 2000.

SU, C.-T. Dynamic vehicle control and scheduling of a multidepot physical distribution system. **Integrated Manufacturing Systems**, v. 10, n. 1, p. 56–65, 1999.

TOTH, P.; VIGO, D. **An Overview of Vehicle Routing Problems**. Society for Industrial and Applied Mathematics, 2002a.

TOTH, P.; VIGO, D. **The vehicle routing problem**. SIAM Monographs on Discrete Mathematics and Applications. Appl. 9, 2002b.

TWINKLE, T.; NAGAVARAPU, S. C.; AZIZIAN, K.; PANDI, R. R; DAUWELS, J. Solving Dial-A-Ride Problems using Multiple Ant Colony System with Fleet Size Minimization. **Advances in Computational Intelligence Systems (ACIS)**, v. 650, p. 325-336, 2018.

WADECO CO LTD. **For iron – torpedo car**. Disponível em <http://www.wadeco.co.jp/en/products/microwave/appli/appli1_2.html>. Acesso em 01/11/2018.

WORLD STEEL ASSOCIATION. **Steel Statistical Yearbook**. Brussels, Belgian, 2018.

XU, H.; PU, P.; DUAN, F. A Hybrid Ant Colony Optimization for Dynamic Multidepot Vehicle Routing Problem. **Discrete Dynamics in Nature and Society**, 2018.

YIN, R. **Metallurgical Process Engineering**. 1 ed. Beijing: Metallurgical Industry Press, 2011.

YU, B.; YANG, Z.; YAO, B. An improved ant colony optimization for vehicle routing problem. **European Journal of Operational Research**, v. 196, n. 1, p. 171–176, 2009.

ZHANGA, Q.; XIONGA, S. Routing optimization of emergency grain distribution vehicles using the immune ant colony optimization algorithm. **Applied Soft Computing**, v. 71, p. 917–925, 2018.

APÊNDICE A – ARTIGO CBSF 2018

Um Sistema Fuzzy para Ajuste Dinâmico dos Parâmetros de um Sistema de Colônia de Formigas Aplicado ao Controle Ferroviário de uma Usina Siderúrgica

Vinícius R. A. Ferreira e Arthur P. de S. Braga

Departamento de Engenharia Elétrica
Universidade Federal do Ceará, Fortaleza, Brasil
viniciusdosreis@hotmail.com, arthurp@dee.ufc.br

Resumo. As meta-heurísticas baseadas em inteligência coletiva, como o algoritmo de otimização por colônia de formigas (*Ant Colony Optimization – ACO*) têm atraído o interesse de muitos pesquisadores devido a sua simplicidade e eficiência na resolução de problemas de otimização. O presente artigo propõe dois sistemas híbridos ACO+Fuzzy que consiste em um método de ajuste dinâmico dos parâmetros do algoritmo. Este ajuste é realizado com a utilização de um sistema Fuzzy para controlar a capacidade de busca local (*exploitation*) e busca global (*exploration*) no espaço de soluções. O cenário das simulações consiste na meta-heurística aplicada a busca do caminho mínimo (*Shortest Path Problem – SPP*) para a otimização de rotas em uma malha ferroviária utilizada para a distribuição de gusa em uma usina siderúrgica. Além do modelamento do grafo e da elaboração do algoritmo, este trabalho faz o comparativo entre de eficácia do algoritmo e aplicabilidade do mesmo em ambiente de produção, através de testes que tentam simular condições reais de operação.

Palavras-chave: Sistema Fuzzy, Ant Colony System, Problema do Caminho Mínimo, Controle Ferroviário, Usina Siderúrgica.

1 Introdução

O trabalho a seguir propõe um método de ajuste dinâmico dos parâmetros de um algoritmo de otimização por colônia de formigas (*Ant Colony Optimization – ACO*) através da utilização de um sistema Fuzzy. Com base no trabalho desenvolvido em [1], este método busca melhorar a eficiência do algoritmo em encontrar o caminho mais curto em uma malha ferroviária, através do controle dos parâmetros que influenciam na capacidade de exploração de novas soluções (*exploration*) e na intensificação das buscas através das soluções já conhecidas (*exploitation*).

Na indústria siderúrgica, é muito comum a utilização de veículos sobre trilhos, como locomotivas e carros torpedos para o transporte de ferro gusa do Alto Forno¹ para

¹ Planta destinada a fundir e reduzir o minério de ferro para transformá-lo em ferro-gusa.

a Aciaria². O tempo gasto nesse percurso influencia diretamente na perda de energia térmica para o meio ambiente, refletindo no maior consumo de redutor no Alto Forno ou na redução da carga metálica de sucata acrescida na Aciaria [2], o que implica em maiores gastos e um conseqüente aumento no custo de produção do aço.

A diminuição deste tempo está relacionada diretamente com a distância percorrida pelo veículo, e é conhecido, em teoria dos grafos, como problema do caminho mínimo (*Shortest Path Problem – SPP*), que consiste na minimização do custo de se percorrer um grafo a partir de um ponto de origem até um determinado nó de destino. Este problema é largamente estudado principalmente nas áreas de Engenharia, Pesquisa Operacional e Ciências da Computação, e possui as mais diversas aplicações.

Existem algoritmos baseados em solução polinomial para a resolução do problema do caminho mínimo, alguns que mais se destacam na literatura são descritos em Bellman-Ford [3] e o Dijkstra [4]. Além disso, diversos estudos trataram a utilização de algoritmos evolucionários utilizando apenas o operador de mutação [5] [6]. Posteriormente, verificou-se que o operador de cruzamento aumentava significativamente a velocidade de convergência da solução para uma solução sub-ótima [7][8]. Entretanto para casos em que o grafo pode sofrer alterações em tempo real, estes algoritmos levam desvantagem frente a utilização de Otimização por Colônia de Formigas (*Ant Colony Optimization – ACO*). Este último apresenta uma característica interessante, que é a possibilidade de ser executado continuamente e se adaptar às mudanças em tempo real. Por este motivo é muito aplicado em problemas de roteamento de redes de comunicação e roteamento de veículos [9].

Alguns trabalhos [10][11] propõem utilização de sistemas Fuzzy para melhorar o desempenho de ACO a partir de mudanças nas taxas de atualização de feromônios. Já em Olivas et al. [1] um sistema Fuzzy é utilizado para parametrização do algoritmo AS_{Rank} aplicado ao Problema do Caixeiro Viajante (PCV). Além da aplicação em ACO, também é possível encontrar a aplicações de lógica Fuzzy para a parametrização de outros algoritmos de inteligência coletiva [12][13].

Os conjuntos Fuzzy e a lógica Fuzzy, propostos por Zadeh [14][15] tornam possível estabelecer relações para grandezas difíceis de mensurar ou que possuam um certo grau de incerteza, inferindo-as em novos fatos através de modelagem linguística. Portanto, com os sistemas Fuzzy, torna-se possível a minimizar a dificuldade de quantificação e a intangibilidade de alguns parâmetros de ajustes de meta-heurísticas, proporcionando a obtenção de melhores resultados.

Este trabalho é estruturado de forma que, na Seção II é feita uma breve introdução ao algoritmo ACO utilizado. Na Seção III, são apresentados os conjuntos Fuzzy. Já na Seção IV detalha-se a metodologia utilizada para a realização das simulações que são apresentadas na Seção V junto aos resultados obtidos. E por fim, na Seção VI é apresentado um relato breve das conclusões.

² Unidade da usina siderúrgica onde o ferro-gusa é convertido em aço.

2 Otimização por Colônia de Formigas

O comportamento das formigas em busca de alimentos, bem como sua sociabilidade, divisão de trabalho e organização em formas de colônias, tem intrigado o ser humano há algumas décadas, existindo relatos de estudos a cerca deste comportamento datados de 1927. Destes estudos e das observações realizadas em sequência, Marco Dorigo [17] propôs o primeiro algoritmo baseado no comportamento exploratório das formigas, chamado de *Ant System* (AS).

Na natureza, uma única formiga é incapaz de se comunicar ou efetivamente buscar comida, mas um grupo de formigas, possuem a habilidade de encontrar e coletar comida de forma eficiente para sua colônia. As formigas se comunicam através de uma substância química chamada feromônio. Ao se locomover, cada formiga deposita uma determinada quantidade dessa substância que pode ser seguida por outras formigas. Cada formiga se desloca de forma aleatória, mas quando encontra um rastro de feromônio, ela pode optar por segui-lo ou não. Caso a formiga decida por seguir o rastro, ela também deposita feromônio pelo trajeto, aumentando ainda mais sua concentração, e, com o aumento da concentração, a probabilidade de outras formigas também escolher o mesmo percurso. Portanto quanto mais formigas andam por um determinado caminho, maior a atratividade deste percurso para as outras formigas [9]. Além disso, quanto menor um caminho até o alimento, mais rápido uma formiga consegue fazer o trajeto de ida e volta (em busca de comida e o retorno até a colônia), e conseqüentemente aumenta a concentração de feromônio neste percurso. Com o passar do tempo, a concentração de feromônio diminui gradativamente devido ao processo de evaporação, o que contribui para que caminhos mais longos se tornem menos atrativos.

A escolha do percurso por uma formiga trata-se de um processo que segue uma distribuição de probabilidades (Eq. 2) que é o elemento chave de funcionamento do algoritmo de otimização por colônia de formigas [18], e possibilita a busca continua por rotas alternativas ao mesmo tempo que oferece certo favorecimento às rotas mais curtas.

Dorigo e Gambardella [19] propuseram o Sistema de Colônia de Formigas (*Ant Colony System – ACS*), como uma melhoria para o AS. Apesar do princípio de funcionamento deste também retratar o comportamento forrageiro das formigas, pode-se destacar algumas diferenças básicas entre o ACS e o AS: uma regra de transição diferente é utilizada para seleção do próximo caminho; a atualização de feromônios foi alterada; e os feromônios passaram a ser atualizados local e globalmente [16].

A regra de escolha do próximo caminho, também conhecida como regra proporcional pseudo-aleatória no ACS, é dada pela equação abaixo:

$$j = \begin{cases} \arg \max_{u \in \mathcal{N}_i^k} \{\tau_{iu} \eta_{iu}^\beta\} & \text{se } q \leq q_0 \\ J & \text{se } q > q_0 \end{cases} \quad (1)$$

onde τ é a concentração de feromônios e η representa a qualidade da aresta; q é um número sorteado randomicamente entre 0 e 1; $q_0 \in [0,1]$ é um parâmetro pré-definido

pelo usuário; $J \in \mathcal{N}_i^k$ representa o próximo nó selecionado aleatoriamente com probabilidade

$$p_{ij}^k = \frac{\tau_{ij}^\alpha \eta_{ij}^\beta}{\sum_{u \in \mathcal{N}_i^k} \tau_{iu}^\alpha + \eta_{iu}^\beta} \quad (2)$$

sendo α parâmetros de influência do feromônio e \mathcal{N}_i^k o conjunto de pontos factíveis a serem visitados a partir do nó i .

Esta regra possibilita um favorecimento do nó com maior concentração de feromônios e de menor tamanho, representado por $\tau_{iu}\eta_{iu}^\beta$, contra o método padrão de tomada de ação também presente no AS. Isso permite que seja feito um ajuste entre a exploração de novos caminhos (busca global) e a intensificação das buscas nos caminhos já conhecidos (busca local).

No ACS, a atualização de feromônios é dada em dois momentos, após a realização do percurso, cada formiga atualiza os feromônios nas arestas percorridas segundo a equação de atualização local (3); a atualização global de feromônios, descrita na equação (4) é realizada ao final da iteração, onde a formiga que realizou o melhor percurso, pode reforçar o depósito de feromônios sobre as arestas deste percurso. E o valor perdido por evaporação é subtraído junto com as atualizações global e local.

$$\tau_{ij} = (1 - \rho_2)\tau_{ij} + \rho_2\tau_0 \quad (3)$$

$$\tau_{ij} = (1 - \rho_1)\tau_{ij} + \rho_1\Delta\tau_{ij}^{(bs)}, \forall (i, j) \in T_{bs} \quad (4)$$

onde ρ_1 e ρ_2 são as taxas de evaporação de feromônios global e local, respectivamente; τ_0 representa um nível mínimo de feromônio; T_{bs} representa a melhor solução encontrada; e

$$\Delta\tau_{ij}^{(bs)} = \begin{cases} \frac{1}{f(x_{bs})} & \text{se } (i, j) \in T_{bs} \\ 0 & \text{caso contrário} \end{cases} \quad (5)$$

sendo $f(x_{bs})$ uma função do tamanho da aresta x ou custo associado a mesma;

No algoritmo proposto com base no ACS, dados um ponto de origem e um de destino, uma quantidade n de formigas são inicializadas no ponto de origem; a partir deste e a cada ponto visitado, obtém-se o próximo nó visitado a partir das equações (1) e (2), excluindo-se os nós já visitados ou os que não estejam disponíveis; após as n formigas terem chegado ao destino calcula-se o custo associado ao percurso, em seguida é realizada a atualização local de feromônios conforme equação (3), e segue para a próxima iteração. A cada iteração verifica-se a melhor solução encontrada e a concentração de feromônio é atualizada (depósito e evaporação) na matriz $T_{53 \times 53}$ conforme equações 3 e 4. O algoritmo chega ao fim quando é atingido o número máximo de iterações.

3 Sistema Fuzzy

Diferentemente da lógica clássica que permite apenas proposições verdadeiras ou falsas, a lógica Fuzzy permite dizer quão pertinente uma determinada proposição é correta ou não. Compartilhando deste princípio, nos sistemas Fuzzy, os elementos de um conjunto possuem graus de associação a este, indicando a certeza/incerteza de o elemento pertencer ao conjunto Fuzzy.

Com as teorias de conjuntos e lógica Fuzzy, é possível modelar controladores, problemas complexos e sistemas não-lineares através do uso da função de pertinência e das regras Fuzzy, que representam as características dos conjuntos Fuzzy e dos sistemas.

Basicamente, controlador Fuzzy é composto por quatro componentes (vide Fig. 1):

1. Regras (ou base de regras): contém as regras Fuzzy que representam o conhecimento ou experiência de um indivíduo sobre o sistema;
2. Fuzzificação: é o processo de conversão das entradas do sistema (variáveis de processo) para graus de pertinência dos conjuntos Fuzzy;
3. Inferência: é o processo de inferência usando as regras e as entradas para geração das saídas Fuzzy;
4. Defuzzificação: fornece um valor representativo do conjunto Fuzzy inferido no passo 3;

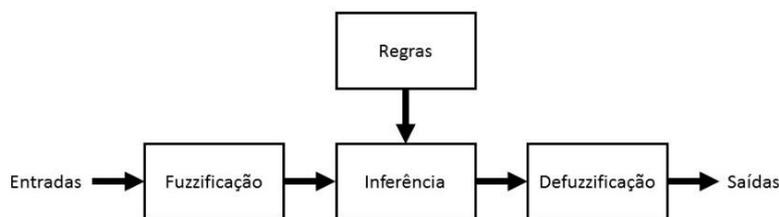


Fig. 1. Arquitetura de um controlador Fuzzy [16].

As duas variáveis que compõem o sistema representam indicadores de desempenho do algoritmo ACS: evolução das iterações e diversidade da solução. Ambas variáveis são percentuais, ou seja, pertencem ao universo dos números reais entre 0 e 1 e foram particionadas em três conjuntos Fuzzy: baixa, média e alta.

Como descrito na Seção IV, foram propostos dois sistemas Fuzzy cada um controlando uma variável de saída. Estas variáveis são dois parâmetros do ACS: α e $q_0 \in [0,1]$. As variáveis de saída foram particionadas em cinco conjuntos: baixa, meio baixa, média, meio alta e alta. As regras do sistema, bem como as funções de pertinência, podem ser vistas na seção seguinte.

Por fim, para o processo de defuzzificação foi adotado o defuzzificador via método do centroide, que é obtido via a média aritmética ponderada pelas pertinências de cada elemento do conjunto Fuzzy.

4 Metodologia

O grafo é uma estrutura formada por nós (também chamados de vértices) e arestas, que são os caminhos que ligam estes nós. Esta estrutura é muito útil na resolução de problemas de roteamento e problemas que envolvam análise combinatória. Para este trabalho, faz-se necessário o modelamento da malha ferroviária representada na figura a seguir em forma de grafo.

A Fig. 2 representa a malha ferroviária simplificada de uma usina siderúrgica. Nela é possível observar os principais pontos de origem e destino dos veículos responsáveis pelo transporte de ferro gusa, material utilizado no processo de fabricação do aço. Os pontos BF, representam as linhas da Alto Forno (*Blast Furnace Plant*) onde o carro torpedo é carregado com ferro gusa. Já os pontos SM são os poços de descarga da Aciaria (*Steel Making Plant*), responsáveis por receber a principal matéria prima para a fabricação do aço.

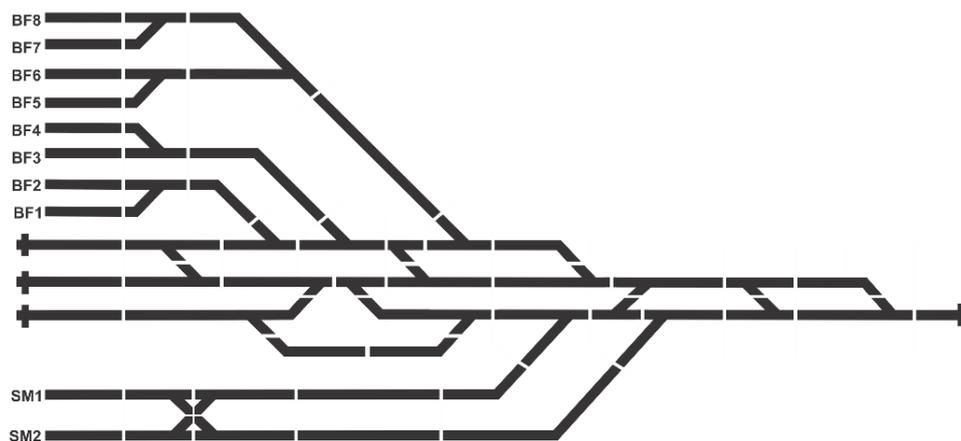


Fig. 2. Malha ferroviária simplificada de uma usina siderúrgica.

O grafo foi modelado como ponderado não direcionado, utilizando-se 53 vértices e 72 arestas. É representado por: dois vetores X e Y correspondentes às coordenadas dos vértices; uma matriz de adjacência $W_{53 \times 53}$ que representa as arestas através da correlação entre dois vértices; uma matriz $D_{53 \times 53}$ contendo o comprimento de cada aresta que representa a distância real (em metros) de cada trecho da malha ferroviária; e uma matriz $L_{53 \times 53}$ representando os trechos não disponíveis.

Este trabalho propõe um controlador Fuzzy tipo Mandani para o ajuste dos parâmetros α e q_0 de um algoritmo ACS com o objetivo de melhorar o desempenho deste na otimização de rotas em uma malha ferroviária. O intuito da comutação destes parâmetros dinamicamente é possibilitar o controle das taxas de busca global e busca local, durante a execução do mesmo. Uma vez que no início do algoritmo, geralmente, deseja-se maior exploração ao longo do espaço de soluções, enquanto que no final, o intuito é concentrar-se nas proximidades da melhor solução já encontrada. Como entradas para o controle serão utilizadas a evolução da quantidade de iterações e a diversidade

das soluções. A quantidade de iterações é medida em percentual já realizado, descrita na equação (6):

$$it(\%) = \frac{nit}{maxIt} \quad (6)$$

onde, nit é o número da iteração atual e $maxIt$ é a quantidade máxima de iterações definida no algoritmo.

Já o indicador de diversidade representa o percentual da quantidade de arestas percorridas por todas as formigas que não fazem parte do conjunto de arestas da melhor solução já encontrada.

$$divers(\%) = \frac{\sum a_{ij}}{\sum b_{ij}} \quad (7)$$

de forma que a_{ij} assume os valores da equação (8) para cada aresta percorrida por qualquer uma das formigas; e b_{ij} representa o número total de arestas percorridas por todas as formigas.

$$a_{ij} = \begin{cases} 1 & \text{se } (i, j) \notin T_{bs} \\ 0 & \text{se } (i, j) \in T_{bs} \end{cases} \quad (8)$$

Ambas as funções de fuzzificação do controlador foram divididas em três conjuntos Fuzzy: Baixo, Médio e Alto. A Fig. 3 mostra a função de pertinência para a evolução das iterações.

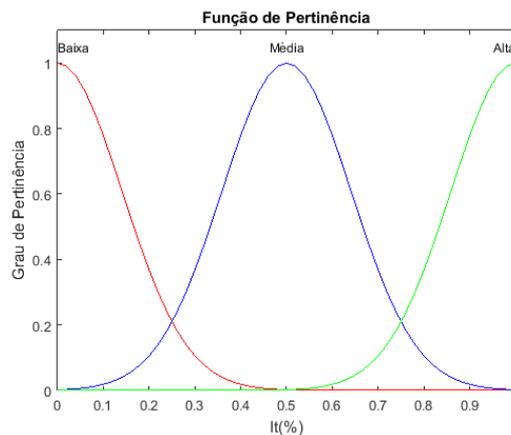


Fig. 3. Função de pertinência para variável de entrada (evolução das iterações).

Também com um formato de Gaussiana, na Fig. 4, tem-se a função de pertinência para a variável que representa a diversidade das soluções.

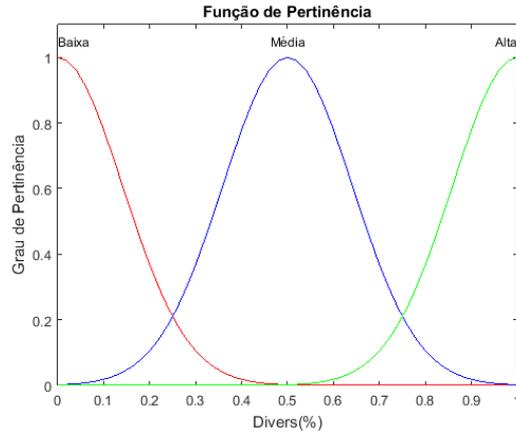


Fig. 4. Função de pertinência para variável de entrada (diversidade das soluções).

Como saída para o controlador, teremos também os mesmos grupos e regras Fuzzy para ambos parâmetros a serem ajustados. São um total de nove regras que relacionam as duas entradas à saída controlada: α e q_0 . As regras são:

1. SE (it é baixa) E (divers é baixa) ENTÃO (saída é baixa)
2. SE (it é baixa) E (divers é média) ENTÃO (saída é meio-baixa)
3. SE (it é baixa) E (divers é alta) ENTÃO (saída é média)
4. SE (it é média) E (divers é baixa) ENTÃO (saída é meio-baixa)
5. SE (it é média) E (divers é média) ENTÃO (saída é média)
6. SE (it é média) E (divers é alta) ENTÃO (saída é meio-alta)
7. SE (it é alta) E (divers é baixa) ENTÃO (saída é média)
8. SE (it é alta) E (divers é média) ENTÃO (saída é meio-alta)
9. SE (it é alta) E (divers é alta) ENTÃO (saída é alta)

A função de pertinência para a saída está dividida em cinco conjuntos Fuzzy de formato triangular: Baixa, Meio Baixa, Média, Meio Alta e Alta, conforme pode ser visto na Fig. 5.

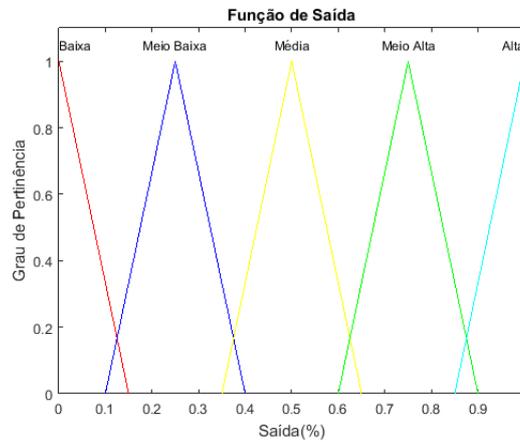


Fig. 5. Função de pertinência para variável de saída (parâmetros α e q_0).

No ACS, tanto o parâmetro q_0 quanto o parâmetro α influenciam de forma semelhante a relação entre busca global e busca local. O primeiro dita a probabilidade da regra de transição, descrita na equação (1), em que, quanto maior q_0 , maior a probabilidade de a formiga escolher o caminho com maior concentração de feromônio e maior qualidade, conseqüentemente, intensificando a busca local pelos caminhos mais explorados. Já o parâmetro α , realça a diferença de concentrações de feromônio na escolha da formiga (vide eq. 2), de forma que, quanto maior α , maior é a probabilidade de a formiga escolher uma aresta com maiores concentrações de feromônios, levando também à diminuição da exploração em favorcimento à busca local.

5 Simulações e Resultados

A fim de testar os controladores propostos, foram realizadas simulações utilizando o grafo da Fig. 2, com quatro trajetos sorteados aleatoriamente, de forma que os trajetos #1, #2, #3 e #4 tiveram como origem os pontos BF7, BF6, SM1 e BF1; e destino, os pontos SM2, BF2, BF4 e SM1, respectivamente. Para cada trajeto, foram realizadas 20 simulações com cada controlador. O algoritmo do ACS proposto por Dorigo [19], assim como os algoritmos melhorados ACS+Fuzzy controlando α ; e ACS+Fuzzy controlando q_0 , foram aplicados aos mesmos trajetos e condições. Os parâmetros utilizados foram obtidos com base nas sugestões dos autores [19] e em alguns testes realizados. Esses parâmetros são apresentados na tabela 1.

Tabela 1. Parâmetros para o ACS.

<i>N. Form.</i>	ρ_1 e ρ_2	α	β	q_0	τ_0
6	0,1	0,5	2	0,5	10^{-5}

Devido às características do grafo, a quantidade de iterações necessárias para a obtenção de resultados satisfatórios é relativamente baixa. Portanto foram realizadas

simulações considerando 20 e 40 iterações, para comparar a influência do número de iterações em cada um dos algoritmos. A utilização de um número fixo de iterações deve-se ao fato de o controle adotado basear-se na evolução das iterações, sendo necessário estimar o percentual de iterações já executadas. Na tabela 2 é possível ver um resumo dos resultados obtidos para 20 iterações. Estes valores são as médias das distâncias (em metros) obtidas por cada algoritmo.

Tabela 2. Resultado médio obtido para 20 iterações.

<i>Trajeto</i>	<i>ACS</i> (m)	<i>ACS + Fuzzy</i> α (m)	<i>ACS + Fuzzy</i> q_0 (m)
#1	5082,1	5104,6	5082,1
#2	3302,5	3302,5	3302,5
#3	4845,8	4845,8	4845,8
#4	5001,5	5001,5	4934,0

Para as simulações com 20 iterações, pode-se observar que o controlador proposto ACS+Fuzzy controlando o parâmetro q_0 , apresentou o melhor resultado na comparação com os demais, com destaque para o trajeto 4, em que o mesmo foi superior. Para os trajetos 2 e 3, tivemos um empate entre os três algoritmos. E, por fim, para o trajeto 1, o ACS+Fuzzy controlando α apresentou o pior desempenho.

Tabela 3. Resultado médio obtido para 40 iterações.

<i>Trajeto</i>	<i>ACS</i> (m)	<i>ACS + Fuzzy</i> α (m)	<i>ACS + Fuzzy</i> q_0 (m)
#1	5082,1	5082,1	5082,1
#2	3302,5	3302,5	3302,5
#3	4845,8	4845,8	4845,8
#4	4956,5	4889,0	4889,0

A tabela 3 apresenta as simulações realizadas para verificar a capacidade de evolução dos algoritmos com a utilização de o dobro de iterações. Para as 40 iterações, é possível ver que os algoritmos híbridos obtiveram resultados iguais, sendo superiores ao ACS, e conforme mostrado na Fig. 6, apresentaram assertividade de 100%, ou seja, em todas as simulações com 40 iterações, estes chegaram a solução ótima conhecida.

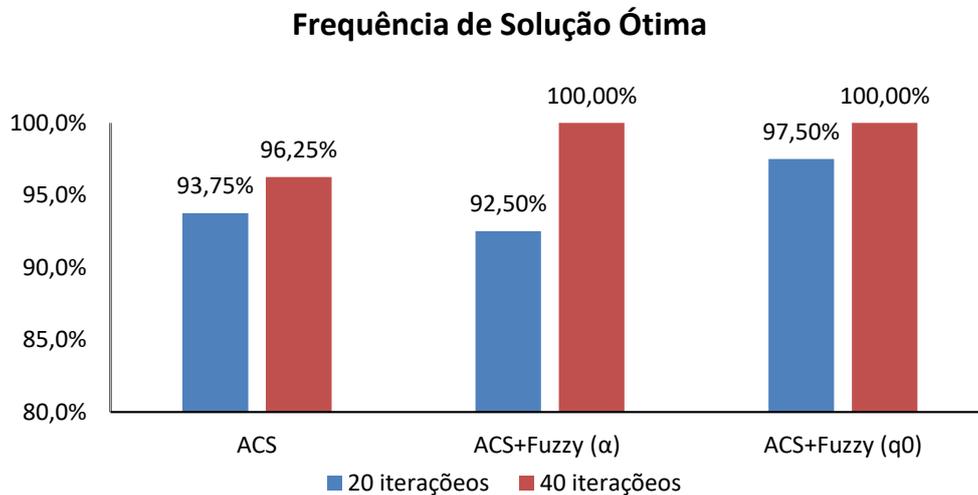


Fig. 6. Frequência que o algoritmo conseguiu obter a solução ótima

Esta figura mostra o percentual de soluções ótimas encontradas para cada um dos algoritmos. É importante observar que com o maior número de iterações, todos os algoritmos apresentaram melhoras significativas, entretanto o melhor resultado obtido em todas as simulações foi com o controlador proposto ACS+Fuzzy controlando o parâmetro q_0 . Além disso, observa-se que para trajetos mais curtos, e conseqüentemente, menos arestas, os três algoritmos obtiveram os mesmos resultados.

6 Conclusões

Com base nos resultados da seção anterior, é possível observar que o controlador fuzzy para ajuste dos parâmetros do ACS resultou em uma melhora significativa frente ao algoritmo ACS original. Apesar deste trabalho não contemplar métricas de esforço computacional, a utilização de um número reduzido de iterações e formigas, indica a viabilidade da aplicação do mesmo em controle de tráfego em tempo real, em que se dispõe de menos tempo para processamento.

Fica como sugestões para trabalhos futuros a comparação dos controladores propostos com outras heurísticas para resolução do problema do caminho mínimo, bem como comparações de esforço computacional.

Referências

1. Olivas, F., Valdez, F. Castillo, O.: A Fuzzy system for parameter adaptation in Ant Colony Optimization. In: Swarm Intelligence (SIS) Symposium. IEEE, Orlando (2014).
2. Magnani, R. et. al.: Redução da Perda Térmica em Carro Torpedo. In Proc. CBECIMAT, pp. 511-518 (2002).
3. Bellman, R., E.: Dynamic Programming. Princeton, NJ: Princeton Univ. Press (1957).

4. Ford, L. R., Fulkerson, D. R.: Flows in Networks. Princeton, NJ: Princeton Univ. Press (1962).
5. Sudholt, D., Thyssen, C.: A Simple Ant Colony Optimizer for Stochastic Shortest Path Problems. *Algorithmica*, vol. 64, pp. 643, (2012).
6. Scharnow, J., Tinnefeld, K., Wegener, I.: The analysis of evolutionary algorithms on sorting and shortest paths problems. *Journal Math. Model. Algorithms* 3(4), 349–366 (2005).
7. Doerr, B., Theile, M.: Improved analysis methods for crossover-based algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'09), pp. 247–254. ACM, New York (2009).
8. Doerr, B., Happ, E., Klein, C.: A tight analysis of the (1+1)-EA for the single source shortest path problem. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC'07), pp. 1890–1895. IEEE Press, New York (2007).
9. Bell, J. E., McMullen, P. R.: Ant colony optimization techniques for the vehicle routing problem. In: *Advanced Engineering Information*. 18, 41–48 (2004).
10. Chang, Y. H., Chang, C. W., Tao, C. W., Lin, H. W., Taur, J. S.: Fuzzy sliding-mode control for ball and beam system with Fuzzy ant colony optimization. In: *Expert Systems with Applications*, vol. 39, no. 3, pp. 3624-3633 (2012).
11. Elloumi, W., Baklouti, N., Abraham, A., Alimi, A. M.: The multi-objective hybridization of particle swarm optimization and Fuzzy ant colony optimization. *Journal of Intelligent and Fuzzy Systems*, vol. 1, IOS Press, Netherlands (2013).
12. Shi, Y., Eberhart, R.: Fuzzy adaptive particle swarm optimization. In: Proceedings of the 2001 Evolutionary Computation Congress, vol. 1, pp. 101-106 (2011).
13. Alfi, A., Fateh, M. M.: Intelligent identification and control using improved Fuzzy particle swarm optimization. In: *Expert Systems with Applications*, vol. 38, no. 10, pp. 12312-12317 (2011).
14. Zadeh, L.: Fuzzy sets. In: *Information & Control*, 8, 338–353 (1965).
15. Zadeh, L.: Fuzzy logic. In: *IEEE Computer Mag.*, vol. 1, pp. 83–93 (1988).
16. Engelbrecht, A. P.: *Computational Intelligence*. 2nd ed. Wiley, South Africa (2007).
17. Dorigo, M., Di Caro, G.: Ant colony optimization: a new metaheuristic. Proceedings of the 1999 Congress on Evolutionary Computation, vol. 2, pp. 1470-1477 (1999).
18. Dorigo, M., Gambardella, L. M.: Ant colonies for the traveling salesman problem. *BioSystem* 43(1), pp. 73–81 (1997).
19. Dorigo, M., Gambardella, L. M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. In: *IEEE Transactions on Evolutionary Computation*, 1(1), pp. 53–66 (1997).